



Redes Neurais para *Data Mining*

Objetivos de aprendizado

- ◆ Entender o conceito e os diferentes tipos de redes neurais artificiais (RNA)
- ◆ Aprender as vantagens e limitações da RNA
- ◆ Entender como as redes neurais de retropropagação aprendem
- ◆ Entender todo o processo de uso das redes neurais
- ◆ Reconhecer a ampla variedade de aplicações de redes neurais

As redes neurais são ferramentas avançadas de data mining usadas quando outras técnicas não produzem modelos preditivos satisfatórios. Como o termo indica, as redes neurais têm uma capacidade de modelagem de inspiração biológica, mas são essencialmente ferramentas de modelagem estatística. Neste capítulo, estudaremos os fundamentos da modelagem de redes neurais, algumas aplicações específicas e o processo de implementação de um projeto de rede neural.

- 6.1 Vinheta de abertura: Usando redes neurais para prever os sabores da cerveja por meio de análise química
- 6.2 Conceitos básicos das redes neurais
- 6.3 Aprendizagem em redes neurais artificiais (RNA)
- 6.4 Desenvolvendo sistemas baseados em redes neurais
- 6.5 Um exemplo de projeto de rede neural
- 6.6 Outros paradigmas das redes neurais
- 6.7 Aplicações de redes neurais artificiais
- 6.8 Demonstração de um software de rede neural

6.1

VINHETA DE ABERTURA: USANDO REDES NEURAIS PARA PREVER OS SABORES DA CERVEJA POR MEIO DE ANÁLISE QUÍMICA

A Coors Brewers Ltd., estabelecida em Burton-upon-Trent, capital da cerveja na Grã-Bretanha, orgulha-se de ter as principais marcas de cerveja do Reino Unido, uma participação de 20% no mercado, anos de experiência e os melhores profissionais no ramo. As marcas populares incluem Carling (a lager mais vendida do país), Grolsch, Coors Fine Light Beer, Sol e Korenwolf.

Problema

O cliente de hoje está diante de uma variedade de opções em termos de bebidas. A escolha de um consumidor depende de diversos fatores, como ânimo, local e ocasião. O objetivo da Coors é assegurar que o cliente escolha sempre uma de suas marcas.

Segundo a empresa, criatividade é a chave para ser bem-sucedido no longo prazo. Para ser a marca escolhida pelo cliente, a Coors precisa ser criativa e previdente quanto ao estado de ânimo do cliente. Uma questão importante relacionada à cerveja é o sabor; cada cerveja tem um sabor próprio. Esses sabores são determinados por meio de análises sensoriais. Essas análises, porém, levam tempo. Se fosse possível reconhecer o sabor da cerveja com base somente na sua composição química, haveria novas possibilidades de criar cervejas que atenderiam às expectativas dos clientes.

A relação entre análise química e sabor da cerveja ainda não foi totalmente entendida. Existem muitos dados sobre sua composição química e análise sensorial. A Coors precisava de um mecanismo que ligasse esses dois itens. As redes neurais foram aplicadas para criar uma ligação entre composição química e análise sensorial.

Solução

Ao longo dos anos, a Coors Brewers Ltd. acumulou uma quantidade significativa de dados relacionados à análise do produto final, completada pelos dados sensoriais fornecidos por analistas internos. Alguns dos dados analíticos de entrada e dos dados sensoriais de saída são mostrados a seguir:

<i>Dados analíticos: entrada</i>	<i>Dados sensoriais: saída</i>
Álcool	Álcool
Cor	Éstere
Amargor calculado	Maltoso
Acetato de etila	Granuloso
Acetato de isobutila	Queimado
Butirato de etila	Lupuloso
Acetato de isoamila	Caramelado
Hexanoato de etila	Adocicado

Uma única rede neural, restrita a uma única qualidade e sabor, foi usada primeiramente para modelar a relação entre os dados analíticos e sensoriais. A rede neural baseava-se em um pacote de soluções fornecido pela NeuroDimension, Inc. (**nd.com**). A rede neural era composta de uma arquitetura tipo perceptron multicamadas (MLP) com duas camadas escondidas. Os dados foram padronizados dentro da rede, com isso possibilitando a comparação entre os resultados dos diversos dados sensoriais de saída. A rede neural foi treinada (para aprender a relação entre os dados de entrada e saída) pela apresentação de muitas combinações de associações relevantes de dados de entrada/saída. Quando não havia melhoria observada no erro da rede nos últimos 100 passos, o treino era automaticamente concluído. O treinamento foi executado 50 vezes a fim de garantir que um erro da rede consideravelmente médio pudesse ser calculado para fins de comparação. Antes da execução de cada treinamento, um diferente conjunto de dados de treinamento e de validação cruzada foi apresentado através da distribuição aleatória dos registros dos dados-fonte, removendo assim qualquer predisposição.

Essa técnica gerou resultados insatisfatórios devido a dois fatores importantes. Primeiro, concentrar-se em uma única qualidade do produto significava que a variação nos dados era bastante baixa. A rede neural não poderia extrair relações úteis dos dados. Segundo, era provável que somente um subconjunto dos dados de entrada fornecidos tivesse um impacto no sabor selecionado de cerveja. O desempenho da rede neural foi afetado pelo “ruído” criado pelos dados de entrada que não tinham impacto no sabor.

Uma variedade de produtos mais diversificada foi incluída no conjunto de treinamento a fim de abranger o primeiro fator. Era mais desafiador identificar os dados analíticos de entrada mais importantes. Esse desafio foi enfrentado através do uso de um *switch* de *software* que permitiu à rede neural ser treinada com todas as combinações possíveis de dados de entrada. O *switch* não foi usado para desativar um dado de entrada significativo; se o dado de entrada significativo fosse desativado, poderíamos esperar que o erro da rede aumentasse. Se o dado de entrada desativado fosse insignificante, então o erro da rede permaneceria inalterado ou seria reduzido devido à remoção do ruído. Essa abordagem é chamada de *busca exaustiva* porque todas as combinações possíveis são avaliadas. Embora conceitualmente simples, a técnica era computacionalmente impraticável com dados de entrada numerosos; o número de combinações possíveis era de 16,7 milhões por sabor.

Era necessário um método mais eficiente de busca de dados de entrada relevantes. Um algoritmo genético era a solução para o problema. Esse algoritmo era capaz de controlar os diferentes *switches* de entrada em resposta à condição de erro da rede neural. O objetivo do algoritmo genético era minimizar a condição de erro da rede. Quando esse mínimo fosse atingido, as configurações do *switch* identificariam os dados analíticos de entrada que estivessem mais propensos a prever o sabor.

Resultados

Após determinar quais dados de entrada eram relevantes, foi possível identificar quais sabores poderiam ser previstos com mais precisão. A rede foi treinada várias vezes usando os dados de entrada relevantes previamente identificados. Antes da execução de cada treinamento, os dados da rede foram randomizados para garantir que um conjunto diferente de dados de treinamento e validação cruzada fosse usado. O erro da rede foi registrado após a execução de cada treinamento. O conjunto de teste usado para avaliar o desempenho da rede treinada continha aproximadamente 80 registros dos dados de amostragem. A rede neural previu com precisão alguns sabores através do uso de dados químicos de entrada. O sabor “queimado” foi previsto com um coeficiente de correlação de 0,87.

Atualmente, um número limitado de sabores é previsto através do uso de dados analíticos. A resposta sensorial é complexa, com muitas interações em potencial e limites de sensibilidade variáveis. A análise instrumental padrão tem parâmetros gerais, e por motivos econômicos e práticos, muitos compostos flavorizantes não são medidos. A relação entre sabor e análise pode ser efetivamente modelada somente se um grande número de analitos que contribuem para o sabor for considerado. Ademais, além das substâncias flavorizantes comuns, contribuintes físicos e de paladar também devem ser considerados no perfil sensorial geral.

Com o desenvolvimento dos parâmetros de entrada, a precisão dos modelos de rede neural será melhorada.

Fontes: C.I. Wilson and L.Threapleton, “Application of Artificial Intelligence for Predicting Beer Flavours from Chemical Analysis,” *Proceedings of the 29th European Brewery Congress*, Dublin, May 17-22, 2003, neurosolutions.com/resources/apps/beer.html (acessado em abril de 2006); R. Nischwitz, M. Goldsmith, M. Lees, P. Rogers and L. MacLeod, “Developing Functional Malt Specifications for Improved Brewing Performance,” *The Regional Institute Ltd.*, regional.org.au/au/abts/1999/nischwitz.htm (acessado em abril de 2006); and coorsbrewers.com (acessado em abril de 2006).

Questões sobre a vinheta de abertura

1. Por que o sabor da cerveja é importante para a rentabilidade da Coors?
2. Qual é o objetivo da rede neural usada na Coors?
3. Por que os resultados iniciais da rede neural da Coors eram insatisfatórios, e o que foi feito para melhorá-los?

4. Quais benefícios a Coors poderá obter caso esse projeto seja bem-sucedido?
5. Quais modificações você apresentaria para melhorar os resultados da predição de sabor da cerveja?

O que podemos aprender com esta vinheta?

Como será visto neste capítulo, as aplicações de redes neurais são usadas em muitas áreas, desde problemas comerciais clássicos de avaliação da capacidade de financiamento de pessoas até aplicações de manufatura, segurança e saúde. Essa vinheta ilustra uma aplicação inovadora em um cenário onde a habilidade humana pode ser considerada a única maneira de avaliar a qualidade. A vinheta mostra que a imaginação de um analista é a única limitação para explorar aplicações de técnicas de *data mining* em geral e de redes neurais *especificamente*. Também aprendemos que, em muitas aplicações da vida real, temos de combinar mais de uma técnica avançada a fim de criar uma aplicação útil. Nessa situação, as redes neurais foram combinadas com algoritmos genéticos, mas outras combinações também são possíveis.

6.2

CONCEITOS BÁSICOS DAS REDES NEURAIS

Redes neurais representam uma metáfora do cérebro para processamento da informação. Esses modelos são biologicamente inspirados e não são uma réplica exata de como o cérebro realmente funciona. Redes neurais têm se revelado sistemas muito promissores em muitas aplicações de previsão e de classificação comercial devido a sua capacidade de “aprender” a partir dos dados, sua natureza não paramétrica (ou seja, sem hipóteses estritas) e sua capacidade de generalizar. **Computação neural** refere-se a uma metodologia de reconhecimento de padrão para aprendizado da máquina. O modelo resultante da computação neural freqüentemente é chamado de **rede neural artificial (RNA)** ou *rede neural*. Redes neurais têm sido usadas em muitas aplicações comerciais para reconhecimento de padrão, previsão, predição e classificação. Computação de rede neural é o principal componente de qualquer conjunto de ferramenta de *data mining* (veja Capítulo 4). Aplicações de redes neurais são muito usadas em finanças, *marketing*, manufatura, operações, sistemas de informação e assim por diante. Por isso, dedicamos este capítulo ao desenvolvimento de uma melhor compreensão dos modelos, métodos e das aplicações de rede neural.

O cérebro humano possui recursos surpreendentes para o processamento da informação e para a resolução de problemas com os quais computadores modernos não conseguem competir em muitos aspectos. Foi pressuposto que um modelo ou sistema que é instruído e auxiliado pelos resultados de pesquisas sobre o cérebro, com uma estrutura similar às redes neurais biológicas, poderia apresentar uma funcionalidade inteligente semelhante. Com base nessa suposição *bottom-up*, a RNA (também conhecida como modelos conexionistas, modelos de processamento paralelo distribuído, sistemas neuromórficos ou simplesmente redes neurais) tem sido desenvolvida como modelos biologicamente inspirados e plausíveis para diversas tarefas.

Redes neurais biológicas são compostas de muitos neurônios biológicos primitivos compactamente interconectados. Cada neurônio possui *axônios* e *dendritos*, projeções semelhantes a dedos que permitem ao neurônio comunicar-se com seus *neurônios* vizinhos através da transmissão e do recebimento de sinais químicos e elétricos. Mais ou menos semelhante à estrutura de seus colegas, a RNA é composta de elementos de processamento simples e interconectados chamados *neurônios artificiais*. No processamento da informação, os elementos de processamento em uma RNA funcionam de maneira simultânea e coletiva em um modo semelhante aos neurônios biológicos. A RNA possui algumas características desejáveis similares àquelas das redes neurais biológicas, como os recursos de aprendizagem, auto-organização e tolerância ao erro.

Avançando em um caminho sinuoso, a RNA é investigada por pesquisadores há mais de meio século. O estudo formal da RNA começou com o trabalho pioneiro de McCulloch e Pitts, em 1943. Encorajados pelos resultados de experimentos e observações biológicas, McCulloch e Pitts (1943) apresentaram um modelo simples de um neurônio artificial binário que capta algumas funções de um neurônio vivo. Considerando as máquinas de processamento de informação como um meio para a modelagem do cérebro, McCulloch e Pitts criaram seu modelo de rede neural usando vários neurônios artificiais binários interconectados. Liderada por um grupo de pesquisadores, a pesquisa sobre redes neurais era muito popular no fim dos anos 1950 e início dos anos 1960. Após uma análise completa de um modelo de rede neural antigo (chamado de **perceptron**, que não usava camada escondida) e também uma avaliação pessimista do potencial da pesquisa por Minsky e Papert, em 1969, o interesse pelas redes neurais diminuiu.

Durante as últimas duas décadas, houve um ressurgimento empolgante nos estudos de RNA devido à introdução de novas topologias de rede, novas funções de ativação e novos algoritmos de aprendizado, além do progresso na neurociência e na ciência cognitiva. De um lado, os avanços em teoria e metodologia superaram muitos obstáculos que atrapalharam a pesquisa sobre redes neurais há algumas décadas. Evidenciadas pelos resultados fascinantes de diversos estudos, as redes neurais estão ganhando aceitação e popularidade. Por outro lado, como solucionadora de problemas complexos, a RNA tem sido usada para resolver numerosos problemas em uma variedade de situações de aplicação. Os recursos desejáveis no processamento de informação neural tornam as redes neurais atraentes para a resolução de problemas complexos. O sucesso inicial das aplicações de rede neural influenciou o interesse renovado da indústria e das empresas.

Redes neurais biológicas e artificiais

O cérebro humano é composto de células especiais chamadas **neurônios**. Essas células não morrem quando uma pessoa é ferida (todas as outras células multiplicam-se para se substituírem e então morrerem). Esse fenômeno pode explicar o porquê de guardarmos informações. O depósito de informações abarca conjuntos de neurônios. Em um cérebro humano, o número estimado de neurônios é de 50 a 150 bilhões, dos quais existem mais de 100 tipos diferentes. Os neurônios são divididos em grupos chamados *redes*. Cada rede contém alguns milhares de neurônios altamente interconectados. Por isso, o cérebro pode ser visto como uma coleção de redes neurais.

A capacidade de aprender e reagir a mudanças em nosso ambiente requer inteligência. O cérebro e o sistema nervoso central controlam o raciocínio e o comportamento inteligente. Pessoas que sofrem danos cerebrais têm dificuldade de aprendizado e reação a ambientes variáveis. Mesmo assim, as partes não danificadas do cérebro podem muitas vezes compensar com novo aprendizado.

Uma parte de uma rede composta de duas células é mostrada na Figura 6.1. A célula inclui em si um **núcleo** (a parte de processamento central da célula). À esquerda da célula 1, os **dendritos** fornecem sinais de entrada para a célula. À direita, o **axônio** envia sinais de saída para a célula 2 através dos terminais do axônio. Esses terminais unem-se aos dendritos da célula 2. Os sinais podem ser transmitidos inalterados, ou podem ser alterados pelas sinapses. Uma **sinapse** é capaz de aumentar ou diminuir a intensidade da ligação entre os neurônios e estimular ou inibir um neurônio subsequente. É aí que a informação é armazenada.

Um modelo de RNA emula uma rede neural biológica. Na verdade, a computação neural usa um conjunto muito limitado de conceitos provenientes dos sistemas neurais biológicos (veja Insights de tecnologia 6.1). É mais uma analogia ao cérebro humano do que um modelo preciso dele. Os conceitos neurais geralmente são implementados como simulações de *software* dos processos paralelos que envolvem os elementos de processamento (também chamados *neurônios artificiais* ou *nós neurais*) interconectados em uma arquitetura de rede. O neurônio artificial recebe sinais de

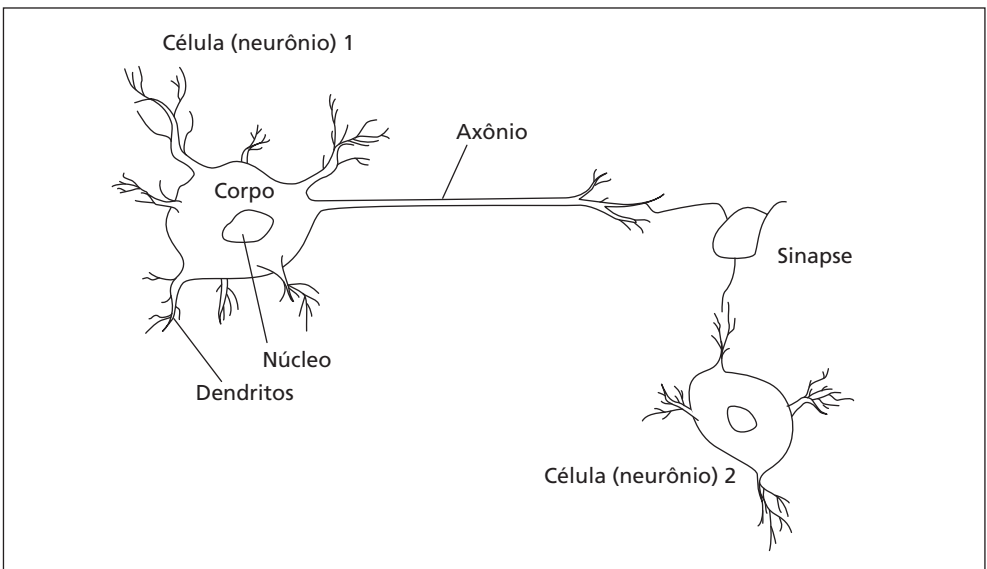


Figura 6.1 Parte de uma rede: duas células biológicas interconectadas.

entrada análogos aos impulsos eletroquímicos que os dendritos dos neurônios biológicos recebem de outros neurônios. Os sinais de saída do neurônio artificial correspondem aos sinais enviados do neurônio biológico através do seu axônio. Esses sinais artificiais podem ser mudados pelos pesos, de maneira semelhante às mudanças físicas que ocorrem nas sinapses (veja Figura 6.2).

Alguns paradigmas de RNA foram propostos para aplicações em vários domínios de problema. Para um exemplo, veja o Caso de Aplicação 6.2. Talvez a maneira mais fácil de diferenciar os diversos modelos seja com base em como esses modelos emulam estruturalmente o cérebro humano, a maneira na qual o modelo neural processa as informações e como os modelos neurais aprendem a executar as tarefas designadas.

Como eles são biologicamente inspirados, os principais elementos de processamento de uma rede neural são neurônios singulares, semelhantes aos neurônios do cérebro. Esses neurônios artificiais recebem a “informação” total de outros neurônios ou estímulos externos de entrada, realizam uma transformação nas entradas e, então, passam a informação transformada para outros neurônios ou estímulos externos de saída. Isso é semelhante a como atualmente se pensa que o cérebro humano funciona. Passar informação de um neurônio para outro pode ser pensado como uma maneira de ativar ou desencadear uma reação de determinados neurônios com base nas informações ou nos estímulos recebidos.

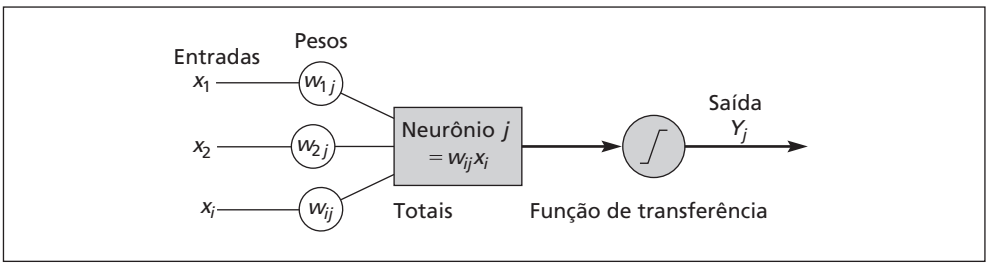


Figura 6.2 Processamento da informação em um neurônio artificial.

INSIGHTS DE TECNOLOGIA 6.1

A relação entre as redes neurais biológicas e artificiais

A lista a seguir mostra algumas das relações entre re-des biológicas e artificiais:

Biológica	Artificial
Corpo	Nó
Dendritos	Entrada
Axônio	Saída
Sinapse	Peso
Velocidade baixa	Velocidade alta
Muitos neurônios (10 ⁹)	Poucos neurônios (dezenas à centenas de milhares)

Zahedi (1993) falou sobre um papel duplo para RNA. Adotamos os conceitos do mundo biológico para melhorar a estrutura dos computadores. A tecnologia de RNA é usada para processamento de informações complexas e inteligência de máquina. Por outro lado, as redes neurais também podem ser usadas como modelos biológicos simples para testar hipóteses sobre processamento de informação neuronal biológico “real”. No contexto de *data mining*, estamos interessados, naturalmente, no uso das redes neurais para aprendizado de máquina e processamento de informação.

Fontes: L. Medsker e J. Liebowitz, *Design and Development of Expert Systems and Neural Networks*, Macmillan, New York, 1994, p.163; e F. Zahedi, *Intelligent Systems for Business: Expert Systems with Neural Networks*, Wadsworth, Belmont, CA, 1993.

Portanto, como a informação é processada por uma rede neural é inerentemente uma função da sua estrutura. As redes neurais podem ter uma ou mais camadas de neurônios. Eles podem estar altamente ou completamente interconectados, ou somente camadas específicas podem estar conectadas. As ligações entre neurônios têm um peso associado. Em essência, o “conhecimento” que a rede possui é condensado nesses pesos de interconexão. Cada neurônio calcula um total ponderado dos valores de entrada do neurônio, transforma essa entrada e repassa seu valor neural como entrada para os neurônios subsequentes. Normalmente, embora não sempre, esse processo de transformação da entrada/saída no nível individual do neurônio é feito de modo não-linear.

CASO DE APLICAÇÃO 6.2

Redes neurais ajudam a reduzir fraude nas telecomunicações

O FIINA (Forum of International Irregular Network Access) estima que as fraudes em telecomunicações resultem em uma perda de US\$55 bilhões por ano em todo o mundo. A maior operadora de telecom da África do Sul estava perdendo mais de US\$37 milhões por ano devido à fraude. A fraude de assinatura – na qual um cliente fornece informações pessoais fraudulentas ou fornece informações pessoais válidas e, em seguida, desaparece – era a maior causa de perda de receitas da empresa. No momento em que a operadora de telecom era alertada sobre a fraude, os fraudadores já tinham mudado para outras vítimas. Outros

tipos de fraude incluem que adulteração e clonagem de cartões telefônicos. Na fraude clipada, o fraudador grampeia a linha telefônica do cliente e, então, vende ligações internacionais por uma fração das taxas normais.

O Minotaur, desenvolvido pela Neural Technologies (neuralt.com), foi implementado para evitar fraudes. Ele usa uma mistura híbrida de sistemas inteligentes e técnicas computacionais tradicionais para fornecer assinatura do cliente e monitoramento de ligação em tempo real para detecção de fraude. Ele processa dados de diversos campos, como registros de dados de evento

(p.ex., *switch*/CDR, SS#7, IPDRs, PIN/autenticação) e dados do cliente (p.ex., fatura e pagamento, ponto de venda, fornecimento), usando uma capacidade de análise *multistream*. As fraudes são detectadas em vários níveis, como, em uma base individual usando conhecimento específico sobre o hábito do assinante, e em uma base global, usando conhecimento genérico sobre o hábito do assinante e padrões de fraude conhecidos. A capacidade neural do Minotaur quer dizer que ele aprende a partir da experiência, fazendo uso de feedback para manter-se atualizado em relação às alterações nos padrões de fraude. Uma combinação de dados de ligação/rede e informações do assinante é delineada e, em seguida, processada, usando técnicas neurais inteligentes, baseadas em casos e regras. Prováveis fraudes são identificadas, reunidas em casos e acompanhadas até sua conclusão por meio de um poderoso e flexível processo operacional baseado em fluxo de trabalho.

Nos primeiros três meses de instalação desse software baseado em rede neural:

- A perda média devido à fraude por caso foi reduzida em 40%.
- O tempo de detecção foi reduzido em 80%.
- O tempo médio necessário para analisar casos suspeitos de fraude foi reduzido em 75%.
- O índice médio de acerto na detecção foi melhorado em 74%.

A combinação de tecnologias neurais, baseadas em casos e em regras, oferece uma taxa de detecção de fraude superior àquela dos sistemas convencionais. Além disso, a capacidade de análise *multistream* o torna extremamente preciso.

*Fontes: Combating Fraud: How a Leading Telecom Company Solved a Growing Problem, neuralt.com/iqs/dlsfa.list/dlcpti.7/downloads.html; A. Shukla, Neural Technologies and Sevis Partner to Eliminate Fraudulent Calls in Fixed and Mobile Networks, February 3, 2006, news.tmcnet.com/news/-neural-sevis-fraud-mobile/2006/02/03/1340423.htm (acessado em abril de 2006); P.A. Estévez, M.H. Claudio, e C.A. Perez, *Prevention in Telecommunications Using Fuzzy Rules and Neural Networks*, cec.uchile.cl/~pestevez/RI0.pdf (acessado em abril de 2006); e *Members and Associate Members Success Stories*, gsm.org/about/membership/success/nt.shtml (acessado em abril de 2006).*

Elementos de RNA

Uma rede neural é composta de elementos de processamento organizados de diferentes maneiras para formar a estrutura da rede. A unidade básica de processamento é o neurônio. Uma série de neurônios está organizada dentro de uma rede. Existem muitas formas de organizar os neurônios; elas são referidas como **topologias**. Uma abordagem popular, conhecida como o paradigma da **retropropagação**, permite que todos os neurônios liguem a saída em uma camada à entrada da camada seguinte, mas não permite qualquer ligação de feedback (Haykin, 1999). Esse é o paradigma mais usado.

Elementos de processamento

Os **elementos de processamento (PE)** de uma RNA são os neurônios artificiais. Cada um dos neurônios recebe entradas, as processa e entrega uma única saída, como mostrado na Figura 6.2. A entrada pode ser dados brutos de entrada ou a saída de outros elementos de processamento. A saída pode ser o resultado final (p.ex., 1 significa sim, 0 significa não) ou pode ser entradas para outros neurônios.

Estrutura da rede

Cada RNA é composta de um conjunto de neurônios, agrupados em camadas. Uma estrutura típica é mostrada na Figura 6.3. Observe as três camadas: *entrada*, *intermediária* (chamada de *camada escondida*) e *saída*. Uma **camada escondida** é uma camada de neurônios que recebe entradas provenientes da camada anterior e as converte em saídas para novo processamento. Diversas camadas escondidas podem ser colocadas entre as camadas de entrada e saída, embora seja comum usar somente uma camada escondida. Nesse caso, a camada escondida converte entradas em uma combinação não-linear e transfere as entradas transformadas para a camada de saída. A interpretação mais comum da camada escondida é como um mecanismo de extração de atributos. Isto é, a camada escondida converte as entradas originais no problema em algumas combinações de alto nível de tais entradas.

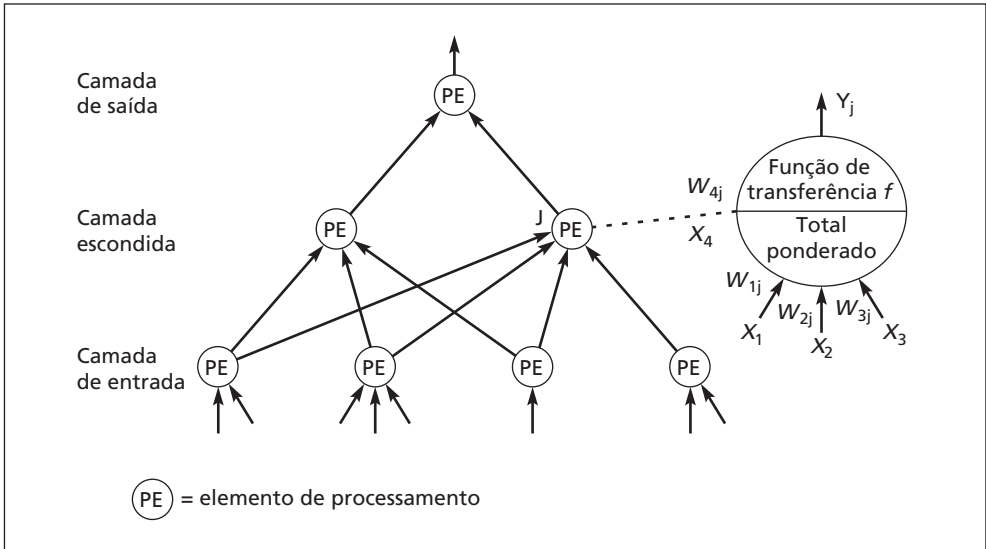


Figura 6.3 Rede neural com uma camada escondida.

Como uma rede biológica, uma RNA pode ser organizada de várias maneiras (ou seja, topologias ou arquiteturas); isto é, os neurônios podem ser interconectados de diferentes formas. Por isso, a RNA aparece em várias configurações chamadas *arquiteturas*. Quando a informação é processada, muitos dos elementos de processamento realizam seus cálculos ao mesmo tempo. Esse **processamento paralelo** assemelha-se ao modo como o cérebro funciona, e difere do processamento serial da computação convencional.

Processamento de informação da rede

Uma vez que a estrutura de uma rede neural é determinada, a informação pode ser processada. Agora, apresentamos os principais conceitos relacionados ao processamento.

Entradas Cada entrada corresponde a um único atributo. Por exemplo, se o problema fosse decidir sobre aprovar ou negar um empréstimo, alguns atributos poderiam ser o índice de rendimento, idade e propriedade imobiliária do requerente. O valor numérico, ou representação, de um atributo é a entrada da rede. Vários tipos de dados, como texto, imagens e voz, podem ser usados como entradas. Talvez seja necessário um pré-processamento para converter os dados em entradas relevantes de dados simbólicos ou graduar os dados.

Saídas As saídas de uma rede contêm a solução para um problema. Por exemplo, no caso de uma solicitação de empréstimo, as saídas podem ser sim ou não. A RNA atribui valores numéricos às saídas, como 1 para sim e 0 para não. O objetivo da rede é calcular os valores da saída. Muitas vezes, é necessário o pós-processamento das saídas porque algumas redes usam duas saídas: uma para sim e outra para não. É comum ter de arredondar as saídas para mais próximo de 0 ou 1.

Pesos de conexão Os **pesos de conexão** são os principais elementos em uma RNA. Eles expressam a intensidade relativa (ou valor matemático) dos dados de entrada ou as muitas conexões que transferem dados de uma camada para outra. Em outras palavras, os pesos expressam a importância relativa de cada entrada para um elemento de processamento e, no final, as saídas. Os pesos são fundamentais já que armazenam os padrões de informação aprendidos. É através de ajustes frequentes dos pesos que uma rede aprende.

Função de soma A **função de soma** calcula os totais ponderados de todos os elementos de entrada que são inseridos em cada elemento de processamento. Uma função de soma multiplica cada valor de entrada pelo seu peso e adiciona os valores para um total ponderado Y . A fórmula para n entradas em um elemento de processamento (veja Figura 6.4a) é:

$$Y = \sum_{i=1}^n X_i W_i$$

Para o j^o neurônio de inúmeros neurônios de processamento em uma camada (veja Figura 6.4b), a fórmula é:

$$Y_j = \sum_{i=1}^n X_i W_{ij}$$

Função de transformação (transferência) A função de soma calcula o estímulo interno, ou nível de ativação, do neurônio. Com base nesse nível, o neurônio pode ou não produzir uma saída. A relação entre o nível de ativação interna e a saída pode ser linear ou não-linear. A relação é expressa por um dos vários tipos de *função de transformação (transferência)*. A **função de transformação (transferência)** combina (ou seja, soma) as entradas vindas de outros neurônios/outras fontes em direção a um neurônio e, em seguida, produz uma saída baseada na

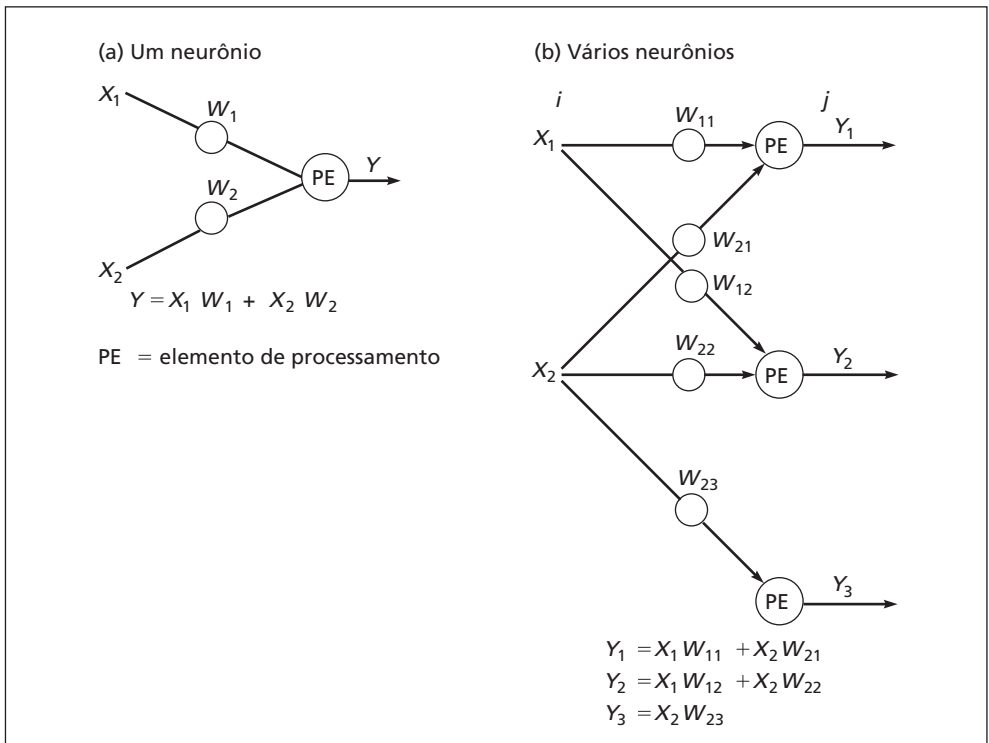


Figura 6.4 Função de soma para um neurônio (a) e vários neurônios (b).

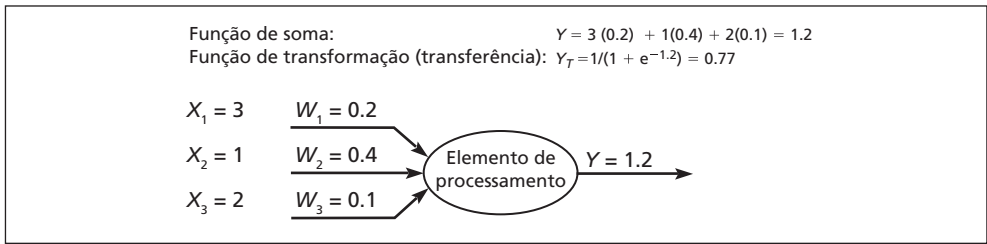


Figura 6.5 Exemplo de funções para RNA.

escolha da função de transferência. A seleção de uma função específica afeta o funcionamento da rede. A **função sigmóide (ativação lógica)** (ou *função de transferência sigmóide*) é uma função de transferência em forma de S com variação de 0 a 1, sendo uma função de transferência não-linear comum e útil:

$$Y_T = 1/(1 + e^{-Y})$$

onde Y_T é o valor transformado (ou seja, padronizado) de Y (veja Figura 6.5).

A transformação modifica os níveis de saída para valores aceitáveis (normalmente entre 0 e 1). Essa transformação é realizada antes que as saídas alcancem o próximo nível. Sem essa transformação, o valor da saída torna-se muito grande, especialmente quando existem diversas camadas de neurônios. Algumas vezes, em vez de uma função de transformação, é usado um *valor limite*. Um **valor limite** é um valor barreira para a saída de um neurônio a fim de ativar o próximo nível de neurônios. Se um valor de saída for menor do que o valor limite, não será passado para o próximo nível de neurônios. Por exemplo, qualquer valor 0,5 ou menor torna-se 0, e qualquer valor acima de 0,5 torna-se 1. Uma transformação pode acontecer na saída de cada elemento de processamento, ou ela pode ser realizada somente nos nós finais de saída.

Camadas escondidas

Aplicações práticas complexas exigem uma ou mais camadas escondidas entre os neurônios de entrada e saída e um número igualmente grande de pesos. Muitas RNAs comerciais incluem três e, algumas vezes, até cinco camadas, cada uma contendo de 10 a 1.000 elementos de processamento. Algumas RNAs experimentais usam milhões de elementos de processamento. Como cada camada aumenta exponencialmente o esforço de treinamento e também aumenta o cálculo necessário, o uso de mais de três camadas escondidas é raro na maioria dos sistemas comerciais.

Arquiteturas da rede neural

Existem vários modelos e algoritmos eficazes de rede neural (consulte Haykin, 1999). Alguns dos mais comuns são *retropropagação (progressivas)*, *memória associativa* e *rede recorrente*. A arquitetura de retropropagação é mostrada na Figura 6.3. As outras duas arquiteturas são mostradas nas Figuras 6.6 e 6.7.

Basicamente, o funcionamento de um modelo completo de rede neural é acionado pela tarefa para a qual foi programado. Por exemplo, modelos de rede neural têm sido usados como classificadores, ferramentas de previsão e otimizadores gerais. Como ainda será mostrado neste capítulo, os classificadores de rede neural normalmente são modelos multicamadas nos quais a informação é passada de uma camada para outra, com o objetivo final de mapear uma entrada

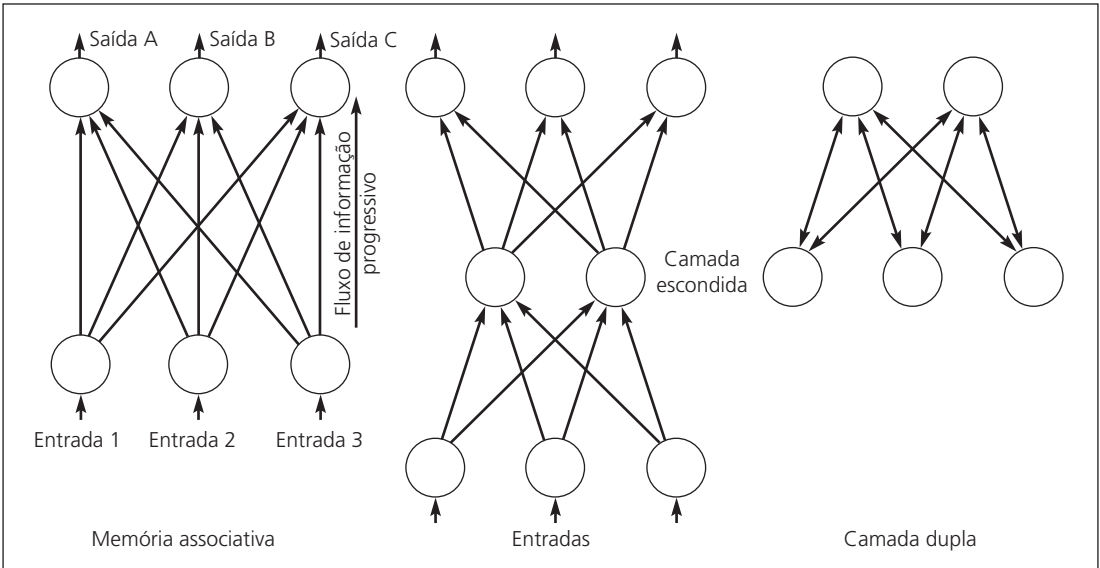


Figura 6.6 Estruturas de rede neural: fluxo progressivo.

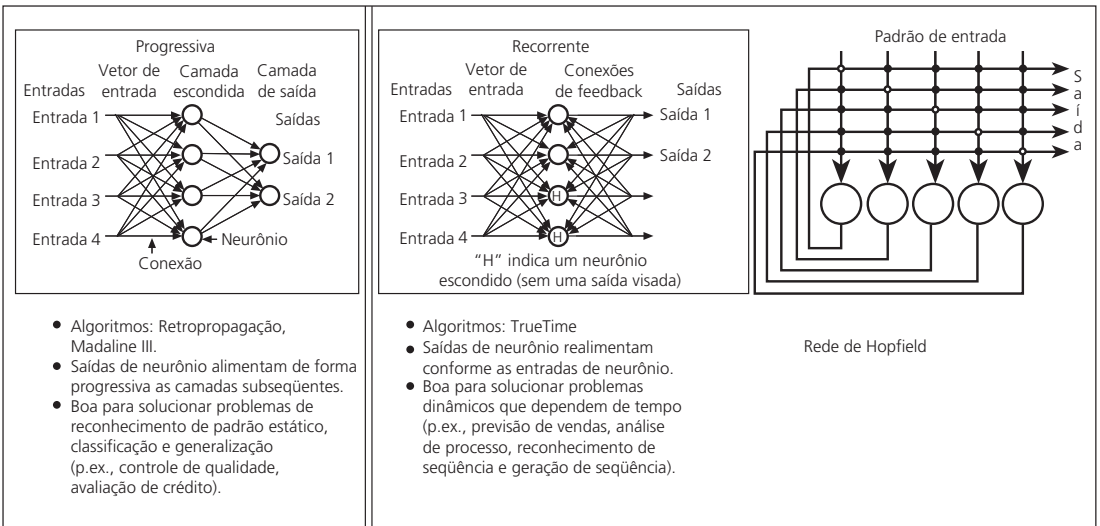


Figura 6.7 Estrutura recorrente comparada com fonte progressiva.

Fonte: Baseado em PC AI, May/June 1992, p.35.

para uma rede para uma categoria específica, conforme identificado pela saída da rede. Por outro lado, um modelo neural usado como otimizador pode ser uma única camada de neurônios, altamente interconectada, e pode calcular valores de neurônio repetidamente até que o modelo convirja a um estado estável. Então, esse estado estável representaria uma solução ideal para o problema sob análise.

Por fim, como uma rede é treinada para executar a tarefa designada é outra característica do modelo identificador. O aprendizado da rede neural pode ocorrer tanto de modo supervisionado como não supervisionado. No aprendizado supervisionado, um conjunto de treinamento amostral é usado para “ensinar” a rede sobre seu domínio de problema. Esse conjunto de treinamento de casos de exemplo (entrada e a saída[s] desejada[s]) é repetidamente apresentado à rede neural. A saída da rede no seu formato atual é calculada e comparada à saída desejada. O **algoritmo de aprendizado** é o procedimento de treinamento usado por uma RNA. O algoritmo de aprendizado usado determina como os pesos de interconexão neural são corrigidos devido a diferenças entre as saídas real e desejada para um membro do conjunto de treinamento. A atualização dos pesos de interconexão da rede continua até que o critério de parada do algoritmo de treinamento seja encontrado (p.ex., todos os casos devem ser classificados corretamente dentro de um determinado nível de tolerância).

Alternativamente, no aprendizado não supervisionado, não há respostas fixadas que a rede tenta aprender. Ao contrário, a rede neural aprende um padrão através de exposição repetida. Portanto, esse tipo de aprendizado pode ser previsto conforme a rede neural adequadamente se auto-organiza ou agrupa seus neurônios relacionados à determinada tarefa desejada.

Redes neurais progressivas com multicamadas são uma classe de modelos que tem potencial em problemas de classificação e previsão. Como o nome indica, esses modelos consistem estruturalmente de múltiplas camadas de neurônios. A informação é passada por intermédio da rede em uma única direção, das camadas de entrada da rede, através de uma ou mais camadas escondidas, em direção à camada de saída dos neurônios. Os neurônios de cada camada estão conectados somente aos neurônios da camada subsequente.

Questões de revisão da seção 6.2

1. O que é uma RNA?
2. Explique os seguintes termos: *neurônio*, *axônio* e *sinapse*.
3. Como funcionam os pesos em uma RNA?
4. Qual é o papel da função de soma?
5. Qual é o papel da função de transformação?

6.3

APRENDIZAGEM EM RNA

Uma importante consideração sobre RNA é o uso de um algoritmo de aprendizado adequado (ou *algoritmo de treinamento*). Algoritmos de aprendizado especificam o processo pelo qual uma rede neural aprende a relação básica entre entrada e saídas, ou apenas entre entradas. Existem centenas delas. Em RNA, os algoritmos de aprendizado também podem ser classificados como aprendizado supervisionado e aprendizado não supervisionado (veja Figura 6.8).

O **Aprendizado supervisionado** usa um conjunto de entradas para as quais as saídas apropriadas (ou seja, desejadas) são conhecidas. Por exemplo, um conjunto histórico de solicitações de empréstimo com êxito ou falha de pagamento do empréstimo por parte do requerente tem um conjunto de parâmetros de entrada e saídas conhecidas presumidas. Em um tipo, a diferença entre as saídas real e desejada é usada para calcular as correções dos pesos da rede neural. Uma variação dessa abordagem simplesmente confirma para cada teste de entrada se a saída está correta à medida que a rede ajusta os pesos em uma tentativa de alcançar os resultados corretos. Exemplos desse tipo de aprendizado são a retropropagação e a rede de Hopfield (Hopfield, 1982). O Caso de Aplicação 6.3 ilustra uma aplicação de aprendizado supervisionado na Microsoft, com o intuito de melhorar a taxa de resposta das correspondências direcionadas a possíveis clientes.

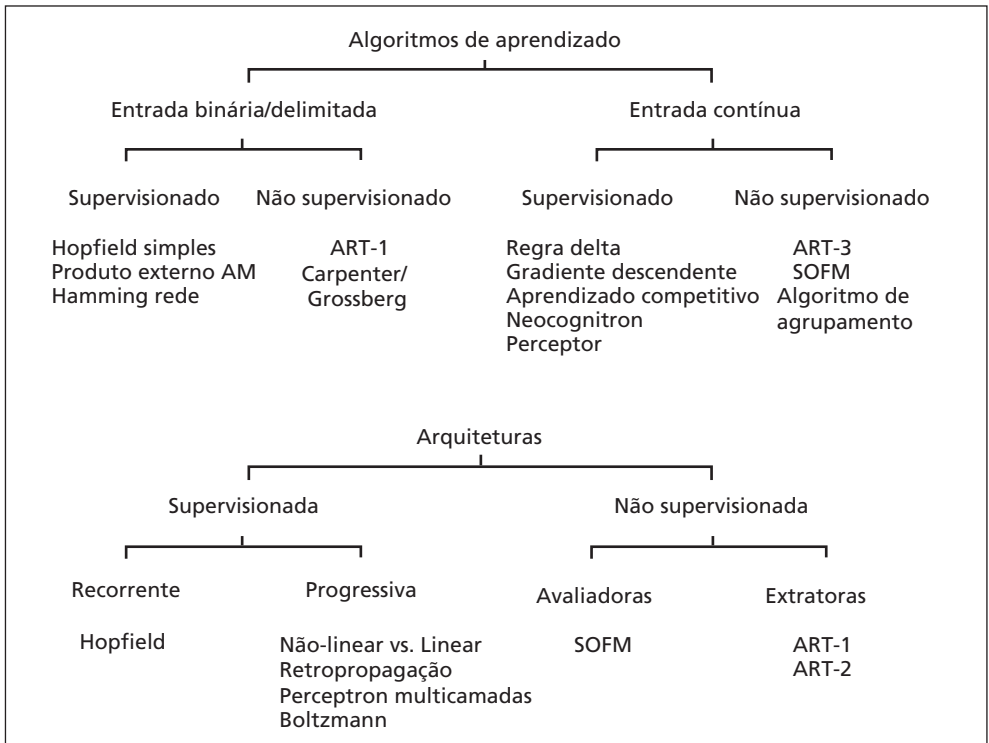


Figura 6.8 Taxonomia de arquiteturas de RNA e algoritmos de aprendizado.

Fonte: Baseado em L. Medsker e J. Liebowitz, *Design and Development of Expert Systems and Neural Computing*, Macmillan, New York, 1994, p.166.

No **aprendizado não supervisionado**, apenas estímulos de entrada são mostrados à rede. A rede é **auto-organizável**; isto é, se organiza internamente de modo que cada elemento de processamento escondido responda estrategicamente a um conjunto diferente de estímulos de entrada (ou grupos de estímulos). Não é fornecido qualquer conhecimento sobre quais classificações (ou

CASO DE APLICAÇÃO 6.3

Redes neurais ajudam a entregar a correspondência da Microsoft ao público almejado

A Microsoft, líder mundial em softwares para computadores, sediada em Redmond, Washington, usa o software de rede neural BrainMaker produzido pela California Scientific (**calsci.com**) para maximizar os rendimentos com mala direta. Todo ano, a Microsoft envia cerca de 40 milhões de itens de mala direta para 8,5 milhões de clientes registrados, com o objetivo de encorajar as pessoas a atualizar seus softwares ou comprar outros produtos relacio-

nados. Em geral, a primeira correspondência inclui todos que estão registrados no banco de dados. O fundamental é direcionar a segunda correspondência somente àqueles mais propensos a responder.

Muitas variáveis foram inseridas na rede neural BrainMaker a fim de obter resultados produtivos. O primeiro passo foi identificar as variáveis relevantes e eliminar as variáveis que não surtiavam efeito. A seguir estão algumas das variáveis significativas:

- Recentidade, calculada em número de dias, que avalia a última vez que algo foi comprado e registrado. Em outras palavras, quanto mais recente for a última compra do cliente, mais chances há de ele/ela comprar mais.
- Primeira data arquivada, que é a data que a pessoa fez sua primeira compra. Isso é uma medida de fidelidade. São altas as chances de um cliente comprar novamente se ele/ela for um(a) cliente fiel.
- O número de produtos comprados e registrados.
- O preço dos produtos comprados e registrados, calculado de acordo com o preço padrão de revenda.
- O número de dias entre a época em que o produto foi lançado e quando foi comprado; pesquisas mostraram que pessoas que tendem a comprar produtos logo que estão disponíveis são os principais indivíduos a serem alcançados.

Outras características pessoais também foram adicionadas e registradas com respostas sim/não.

Antes do treinamento, a informação obtida a partir das respostas dos clientes foi inserida em um formato que pudesse ser usado pela rede, e respostas sim/não foram transformadas em dados numéricos.

Mínimos e máximos foram determinados em variáveis específicas.

Inicialmente, a rede foi treinada com 25 variáveis. Os dados foram retirados de sete ou oito campanhas a fim de torná-los sortidos e representarem todos os aspectos do negócio, incluindo os pontos de vista de Windows e Mac, de produtos de nível de preço alto e baixo.

Antes de a Microsoft começar a usar o BrainMaker, uma correspondência comum obteria uma taxa de resposta de 4,9%. Com o uso do BrainMaker, a taxa de resposta aumentou para 8,2%. A rede neural foi testada com dados de 20 campanhas diferentes, não sendo usadas as respostas conhecidas durante o treinamento. Os resultados mostraram economias consistentes e frequentes. Uma correspondência comum resultou em uma redução de custos de 35% para a Microsoft.

Fontes: California Scientific, “Maximize Returns on Direct Mail with BrainMaker Neural Networks Software”, calsci.com/DirectMail.html; e G. Piatetsky-Shapiro, *ISR: Microsoft Success Using Neural Network for Direct Marketing*, Kdnuggets.com/news/94/n9.txt (acessado em março de 2006).

seja, saídas) são corretas, e aquelas que a rede infere podem ou não ser significativas para o desenvolvedor da rede (isso é útil para análise de *cluster*). Entretanto, ao determinar parâmetros de modelo, podemos controlar o número de categorias nas quais uma rede classifica as entradas. De qualquer maneira, um ser humano deve examinar as categorias finais a fim de atribuir significado e determinar a utilidade dos resultados. Exemplos desse tipo de aprendizado são a **teoria da ressonância adaptativa (ART)** (ou seja, uma arquitetura de rede neural que tem como objetivo ser semelhante ao cérebro em modo não supervisionado) e os **mapas de características auto-organizáveis de Kohonen** (ou seja, modelos de rede neural para aprendizado da máquina).

Como mencionado anteriormente, muitos paradigmas de rede neural foram propostos para vários domínios de tomada de decisão. Um modelo neural adequado para problemas de classificação (p.ex., predição de falência) é o MLP progressivo. Redes multicamadas avaliam continuamente os neurônios (ou seja, elementos de processamento), são treinadas de modo supervisionado e são compostas de uma ou mais camadas de nós (ou seja, nós escondidos) entre os nós de entrada e saída. Uma rede neural progressiva típica é mostrada na Figura 6.3. Os nós de entrada representam onde a informação é apresentada à rede, os nós de saída fornecem a “decisão” da rede neural, e os nós escondidos, por meio dos pesos de interconexão, contêm o mapeamento adequado das entradas até as saídas (ou seja, decisões).

O algoritmo de aprendizado de retropropagação é a maneira padrão de implementar um treinamento supervisionado para redes neurais progressivas. É uma técnica iterativa do gradiente descendente planejada para minimizar uma função de erro entre a saída real da rede e sua saída desejada, conforme especificado no conjunto de dados de treinamento. O ajuste dos pesos de interconexão, que contêm a função de mapeamento propriamente dita, começa no nó de saída onde a medida de erro é inicialmente calculada e, então, propagada de volta através das camadas da rede, em direção a camada de entrada. Maiores detalhes estão incluídos na próxima seção.

O processo de aprendizagem geral da RNA

No aprendizado supervisionado, o processo de aprendizagem é indutivo; isto é, os pesos de conexão são obtidos de casos existentes. O processo de aprendizagem normal envolve três tarefas (veja Figura 6.9):

1. Calcular saídas temporárias.
2. Comparar as saídas com as metas desejadas.
3. Ajustar os pesos e repetir o processo.

Quando saídas existentes estão disponíveis para comparação, o processo de aprendizagem começa pela determinação dos pesos de conexão, através de regras ou aleatoriamente. A diferença entre a saída real (Y ou Y_T) e a saída desejada (Z) para um determinado conjunto de entradas é um erro chamado *delta* (em cálculo infinitesimal, o símbolo grego delta, Δ , significa “diferença”).

O objetivo é minimizar o delta (ou seja, reduzi-lo a 0 se possível), o que é feito pelo ajuste dos pesos da rede. A solução é alterar os pesos na direção certa, fazendo mudanças que reduzam o delta (ou seja, erro). Mostraremos como isso é feito mais adiante.

O processamento de informação com uma RNA consiste em uma tentativa de reconhecer padrões de atividades (ou seja, **reconhecimento de padrão**). Durante as fases de aprendizagem, os pesos de interconexão mudam em resposta aos dados de treinamento apresentados ao sistema.

RNAs diferentes calculam o delta de maneiras diferentes, dependendo do algoritmo de aprendizado usado. Existem centenas de algoritmos de aprendizado para diversas situações e configurações; alguns deles serão discutidos ainda neste capítulo.

Como a rede aprende

Imagine um único neurônio que aprende a operação inclusiva OU – um problema clássico em lógica matemática. Existem dois elementos de entrada, X_1 e X_2 . Se um ou outro ou ambos possuem um valor positivo, o resultado também é positivo. Isso pode ser mostrado como segue:

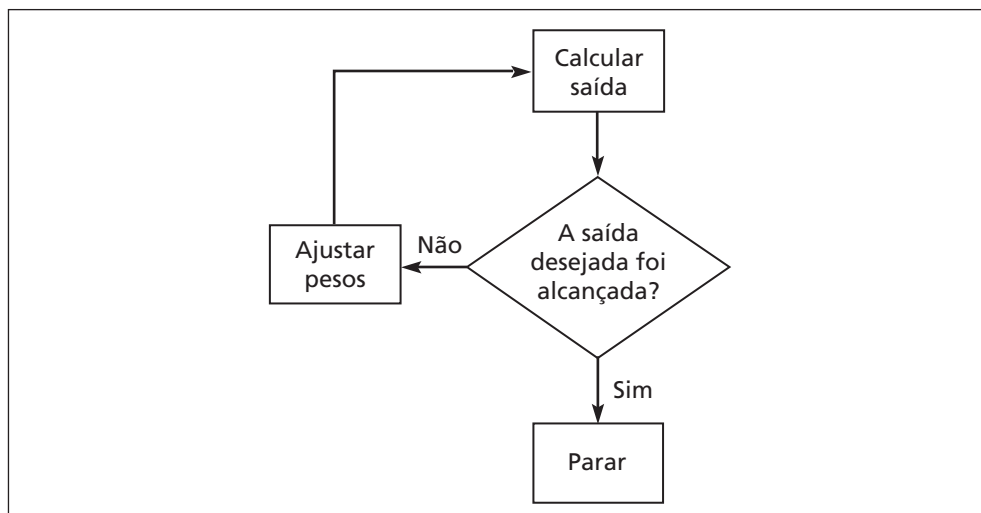


Figura 6.9 Processo de aprendizagem de uma RNA.

Entradas			
Casos	X_1	X_2	Resultados desejados
1	0	0	0
2	0	1	1 (positivo)
3	1	0	1 (positivo)
4	1	1	1 (positivo)

O neurônio deve ser treinado para reconhecer os padrões de entrada e classificá-los de modo a fornecer as saídas correspondentes. O procedimento é apresentar ao neurônio a sequência de quatro padrões de entrada fazendo com que os pesos sejam ajustados após cada repetição (usando o *feedback* do erro encontrado através da comparação da estimativa com o resultado desejado). Esse passo é repetido até que os pesos convirjam a um conjunto uniforme de valores que permitam aos neurônios classificar corretamente cada uma das quatro entradas. Os resultados mostrados na Tabela 6.1 foram gerados no Excel. Nesse exemplo simples, uma função limite é usada para estimar a soma dos valores de entrada. Após calcular as saídas, uma medida do erro (ou seja, delta) entre a saída e os valores desejados é usada para atualizar os pesos, reforçando posteriormente os resultados corretos. Em qualquer passo do processo para um neurônio j , temos:

$$\text{delta} = Z_j - Y_j$$

onde Z e Y são, respectivamente, as saídas desejada e real. Portanto, os pesos atualizados são:

$$W_i(\text{final}) = W_i(\text{inicial}) + \text{alfa} \times \text{delta} \times X_i$$

onde alfa é um parâmetro que controla quão rápido o aprendizado acontece. Ele é chamado de **taxa de aprendizado**. A escolha do parâmetro taxa de aprendizado pode ter um impacto em quão

Tabela 6.1 Exemplo de aprendizado supervisionado

Passo	Inicial							Final	
	X_1	X_2	Z	W_1	W_2	Y	Delta	W_1	W_2
1	0	0	0	0,1	0,3	0	0,0	0,1	0,3
	0	1	1	0,1	0,3	0	1,0	0,1	0,5
	1	0	1	0,1	0,5	0	1,0	0,3	0,5
	1	1	1	0,3	0,5	1	0,0	0,3	0,5
2	0	0	0	0,3	0,5	0	0,0	0,3	0,5
	0	1	1	0,3	0,5	0	0,0	0,3	0,7
	1	0	1	0,3	0,7	0	1,0	0,5	0,7
	1	1	1	0,5	0,7	1	0,0	0,5	0,7
3	0	0	0	0,5	0,7	0	0,0	0,5	0,7
	0	1	1	0,5	0,7	0	0,0	0,5	0,7
	1	0	1	0,5	0,7	0	1,0	0,7	0,7
	1	1	1	0,7	0,7	1	0,0	0,7	0,7
4	0	0	0	0,7	0,7	0	0,0	0,7	0,7
	0	1	1	0,7	0,7	0	0,0	0,7	0,7
	1	0	1	0,7	0,7	0	0,0	0,7	0,7
	1	1	1	0,7	0,7	1	0,0	0,7	0,7

^aParâmetros: alfa = 0,2; limite = 0,5, a saída é zero se o total ($W_1 * X_1 + W_2 * X_2$) não for maior do que 0,5.

rápido (ou quão corretamente) uma rede neural aprende. Um valor alto para a taxa de aprendizado pode levar a muitas correções nos valores de peso, resultando em ir e voltar dentre os valores de peso possíveis e nunca atingir o ideal, que pode estar em algum ponto intermediário das extremidades. Uma taxa de aprendizado muito baixa pode desacelerar o processo de aprendizagem. Na prática, um analista de rede neural deve tentar usar muitas opções diferentes de taxas de aprendizado para alcançar o aprendizado ideal.

A maioria das implementações do processo de aprendizagem também inclui um parâmetro de contrabalanceamento chamado *momentum* para fornecer um equilíbrio à taxa de aprendizado. Essencialmente, enquanto a taxa de aprendizado procura corrigir o erro, o **momentum** procura desacelerar o aprendizado. Atualmente, muitos dos programas de software disponíveis para redes neurais podem selecionar automaticamente esses parâmetros para o usuário ou deixar o usuário experimentar várias combinações diferentes desses parâmetros.

Como mostrado na Tabela 6.1, cada cálculo usa um dos pares X_1 e X_2 e o valor correspondente para a operação OU, junto com os valores iniciais W_1 e W_2 dos pesos do neurônio. Inicialmente, os pesos são determinados como valores aleatórios, e a taxa de aprendizado, α , é fixada baixa. O delta é usado para extrair os pesos finais, que então se tornam os pesos iniciais na próxima repetição (ou seja, linha).

Os valores iniciais dos pesos para cada entrada são transformados usando a equação mostrada anteriormente a fim de determinar os valores que serão usados com a próxima entrada (ou seja, linha). O valor limite (0,5) fixa a saída Y para 1 na próxima linha se o total ponderado de entradas for maior do que 0,5; caso contrário, Y é fixado para 0. No primeiro passo, duas das quatro saídas estão incorretas ($\text{delta} = 1$), e um conjunto homogêneo de pesos não foi encontrado. Nos passos subsequentes, o algoritmo de aprendizado melhora os resultados, até que finalmente cria um conjunto de pesos que fornece os resultados corretos ($W_1 = W_2 = 0,7$ no passo 4 da Tabela 6.1). Uma vez determinado, um neurônio com esses valores de peso pode realizar a operação OU rapidamente.

Ao desenvolver uma RNA, é feita uma tentativa de encaixar a característica do problema em um dos algoritmos de aprendizado conhecidos. Existem programas de software para todos os diferentes algoritmos, como a retropropagação, que descreveremos a seguir. Existem muitas variantes desse algoritmo, mas as noções centrais de todos eles são semelhantes.

Retropropagação

Retropropagação (reduzido de *propagação retrógrada do erro*) é o algoritmo de aprendizado supervisionado mais usado na computação neural (Principe et al., 2000) e é muito fácil de implementar. Uma rede de retropropagação inclui uma ou mais camadas escondidas. Esse tipo de rede é considerado progressivo porque não há interconexões entre a saída de um elemento de processamento e a entrada de um nó na mesma camada ou em uma camada anterior. Padrões corretos externamente fornecidos são comparados à saída da rede neural durante o treinamento (supervisionado), e o *feedback* é usado para ajustar os pesos até que a rede tenha categorizado todos os padrões de treinamento o mais corretamente possível (a tolerância ao erro é determinada com antecedência).

Começando com a camada de saída, os erros entre as saídas real e desejada são usados para corrigir os pesos das conexões da camada anterior (veja Figura 6.10). Para qualquer neurônio de saída j , o erro (delta) = $(Z_j - Y_j)(df/dx)$, onde Z e Y são, respectivamente, as saídas desejada e real. Usar a função sigmóide, $f = [1 + \exp(-x)]^{-1}$, onde x é proporcional ao total das entradas ponderadas para o neurônio, é uma maneira eficaz de calcular a saída de um neurônio na prática. Com essa função, a derivada da função sigmóide $df/dx = f(1 - f)$ e o erro são uma simples função das saídas

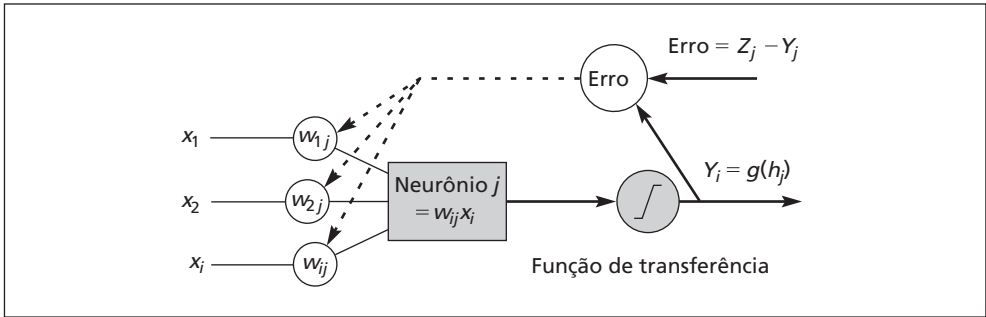


Figura 6.10 Retropropagação de erros para um único neurônio.

desejada e real. O fator $f(1 - f)$ é a *função logística*, que serve para manter a correção do erro bem delimitada. Os pesos de cada entrada para o j^o neurônio são, então, mudadas em proporção ao erro calculado. Uma expressão mais complicada pode ser obtida para trabalhar retroativamente de maneira semelhante a partir dos neurônios de saída através das camadas escondidas de modo a calcular as correções para os pesos associados dos neurônios internos. Esse método complicado é uma abordagem repetitiva para a resolução de um problema de otimização não-linear que é muito parecido em definição àquele que caracteriza a regressão multilinear.

O algoritmo de aprendizado inclui os seguintes procedimentos:

1. Iniciar pesos com valores aleatórios e determinar outros parâmetros.
2. Ler o vetor de entrada e a saída desejada.
3. Estimar a saída real por meio de cálculos, trabalhando progressivamente através das camadas.
4. Calcular o erro.
5. Alterar os pesos através do trabalho progressivo, da camada de saída passando pelas camadas escondidas.

Esse procedimento é repetido para todo o conjunto de vetores de entrada até que as saídas desejada e real harmonizem com qualquer tolerância predeterminada. Dado os requisitos de cálculo para uma repetição, uma rede grande pode levar muito tempo para ser treinada; por isso, em uma variação, um conjunto de casos é executado progressivamente e um erro agregado é inserido retroativamente para acelerar o aprendizado. Algumas vezes, dependendo dos pesos aleatórios iniciais e dos parâmetros de rede, a rede converge a um nível de desempenho satisfatório. Quando esse é o caso, novos pesos aleatórios devem ser criados, e os parâmetros de rede, ou mesmo sua estrutura, devem ser modificados antes que outra tentativa seja feita. Pesquisas atuais buscam desenvolver algoritmos e usar computadores em paralelo para melhorar esse processo. Por exemplo, algoritmos genéticos podem ser usados para orientar a escolha da estrutura de rede, como mencionado na vinheta de abertura.

Questões de revisão da Seção 6.3

1. Faça uma breve descrição da retropropagação.
2. Qual é a finalidade de um valor limite em um algoritmo de aprendizado?
3. Qual é a finalidade de uma taxa de aprendizado?
4. Como o erro entre os resultados real e previsto afeta o valor dos pesos nas redes neurais?

DESENVOLVENDO SISTEMAS BASEADOS EM REDES NEURAIS

Embora o processo de desenvolvimento da RNA seja semelhante às metodologias estruturadas de projeto dos tradicionais sistemas de informação baseados em computador, algumas fases têm alguns aspectos singulares. No processo aqui descrito, supomos que as etapas preliminares para o desenvolvimento de sistemas, como determinação de requisitos de informação, realização de uma análise de viabilidade e obtenção de um líder para o projeto na alta gerência, foram completadas com sucesso. Essas etapas são gerais para qualquer sistema de informação.

Como mostrado na Figura 6.11, o processo de desenvolvimento para uma aplicação de RNA inclui nove etapas. Na etapa 1, os dados a serem usados para treinamento e teste da rede são coletados. Duas considerações importantes: o problema específico deve ser acessível à solução de rede neural; e, dados adequados existem e podem ser obtidos. Na etapa 2, os dados de treinamento devem ser identificados, e um plano deve ser feito para testar o desempenho da rede.

Nas etapas 3 e 4, uma arquitetura de rede e um método de aprendizagem são selecionados. A disponibilidade de uma determinada ferramenta de desenvolvimento ou a habilidade do pessoal do desenvolvimento deve determinar o tipo de rede neural a ser criada. Além disso, certos tipos de problema demonstraram altos índices de sucesso com determinadas configurações (p.ex., redes neurais progressivas com multicamadas para predição de falência, conforme descrito na próxima seção). Aqui, as considerações importantes são o número exato de neurônios e camadas. Alguns pacotes usam algoritmos genéticos para selecionar o desenho da rede.

Existem parâmetros para ajustar a rede de acordo com o nível desejado de desempenho do aprendizado. Parte do processo na etapa 5 é a inicialização dos pesos e parâmetros de rede, seguidos pela modificação dos parâmetros à medida que o *feedback* sobre desempenho do aprendizado é recebido. Muitas vezes, os valores iniciais são importantes na determinação da eficiência e duração do treinamento. Alguns métodos alteram os parâmetros durante o treinamento para aprimorar o desempenho.

A etapa 6 transforma os dados de aplicação no tipo e no formato exigidos pela rede neural. Isso pode exigir a gravação de um *software* para pré-processar os dados ou realizar essas operações diretamente em um pacote de RNA. Técnicas e processos de armazenamento e manipulação de dados devem ser projetados para retreinamento prático e eficiente da rede neural, quando necessário. Não raro, a apresentação e disposição dos dados de aplicação influenciam a eficiência e, possivelmente, a precisão dos resultados.

Nas etapas 7 e 8, treinamento e teste são conduzidos repetidamente por meio da exposição à rede dos dados de entrada e dados de saída desejados ou conhecidos. A rede calcula as saídas e ajusta os pesos até que as saídas calculadas estejam dentro de uma tolerância aceitável das saídas conhecidas para os casos de entrada. As saídas desejadas e suas relações com os dados de entrada são obtidas de dados históricos (ou seja, uma parcela dos dados coletados na etapa 1).

Na etapa 9, um conjunto estável de pesos é obtido. Agora, a rede pode reproduzir as saídas desejadas, dado as entradas como aquelas do conjunto de treinamento. A rede está pronta para ser usada como um sistema independente ou como parte de outro sistema de *software* no qual novos dados de entrada lhe serão apresentados e suas saídas serão uma decisão recomendada.

Nas próximas seções, examinaremos essas etapas mais detalhadamente.

Coleta e preparação dos dados

As primeiras duas etapas no processo de desenvolvimento da RNA envolvem coletar os dados, separá-los em um conjunto de treinamento e um de teste. Os casos de treinamento são usados para

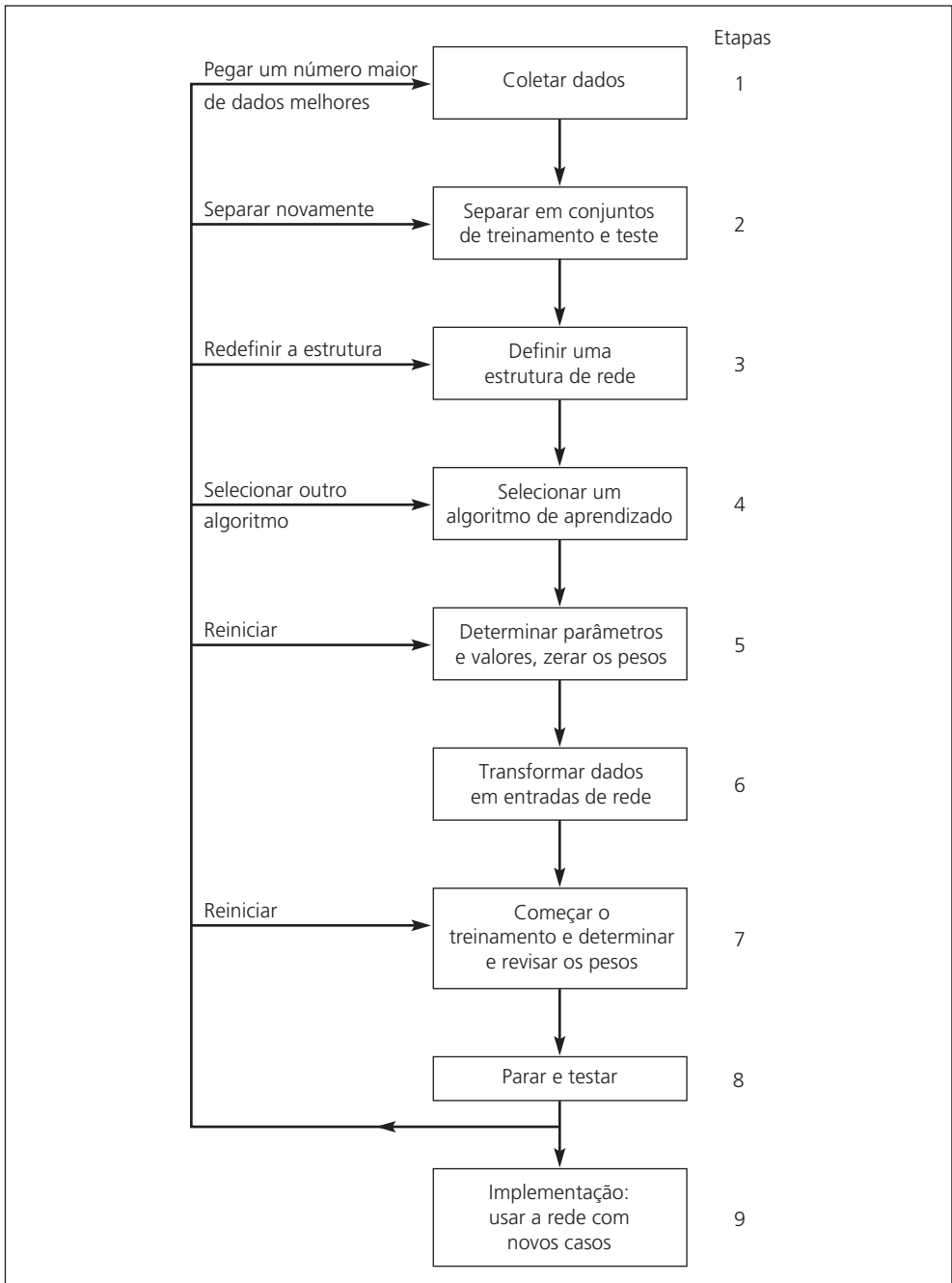


Figura 6.11 Fluxograma do processo de desenvolvimento de uma RNA.

ajustar os pesos, e os casos de teste são usados para validação da rede. Os dados usados para treinamento e teste devem incluir todos os atributos úteis para resolução do problema. O sistema pode aprender apenas tanto quanto os dados podem dizer. Portanto, a coleta e preparação dos dados são a etapa mais importante na criação de um sistema eficiente.

Em geral, quanto mais dados forem usados melhor. Enormes conjuntos de dados aumentam o tempo de processamento durante o treinamento, mas melhoram a precisão do treinamento e muitas vezes levam a uma convergência mais rápida em um conjunto satisfatório de pesos. Para um conjunto de dados médio, 80% dos dados são selecionados aleatoriamente para treinamento e 20% são usados para teste; para conjuntos pequenos de dados, todos os dados são usados para treinamento e teste; e para conjuntos grandes de dados, uma amostra suficientemente ampla é extraída e tratada como um conjunto de dados médio.

Por exemplo, digamos que um banco quer criar um sistema baseado em rede neural a fim de usar os dados financeiros dos clientes para determinar se eles podem falir. Os bancos precisam, primeiro, identificar quais dados financeiros devem ser usados como entradas e de que forma obtê-los. Cinco atributos podem ser entradas úteis: (1) capital de giro/total de ativos, (2) lucros acumulados/total de ativos, (3) lucro antes de juros e imposto de renda/total de ativos, (4) valor de mercado do patrimônio/dívida total, e (5) vendas/total de vendas. A saída é uma variável binária: falência ou não.

Escolha da estrutura de rede

Após os conjuntos de dados de treinamento e teste serem identificados, a próxima etapa é projetar a estrutura das redes neurais. Isso inclui a escolha de uma topologia e a determinação de (1) nós de entrada, (2) nós de saída, (3) número de camadas escondidas e (4) número de nós escondidos. A topologia progressiva com multicamadas frequentemente é usada em aplicações comerciais, embora outros modelos também estejam começando a descobrir algum uso comercial.

O esquema dos nós de entrada deve ser baseado nos atributos do conjunto de dados. No exemplo da predição de falência, o banco deveria escolher uma estrutura de três camadas que incluísse uma camada de entrada, uma camada de saída e uma camada escondida, por exemplo. A camada de entrada contém cinco nós, sendo cada um deles uma variável, e a camada de saída contém um nó com 0 para falência e 1 para seguro. Determinar o número de nós escondidos é delicado. Algumas heurísticas foram propostas, mas nenhuma delas é indiscutivelmente a melhor. Uma abordagem típica é escolher o número médio de nós de entrada e saída. No caso anterior, o nó escondido poderia ser definido para $(5 + 1)/2 = 3$. A Figura 6.12 mostra uma possível estrutura para o problema de predição de falência.

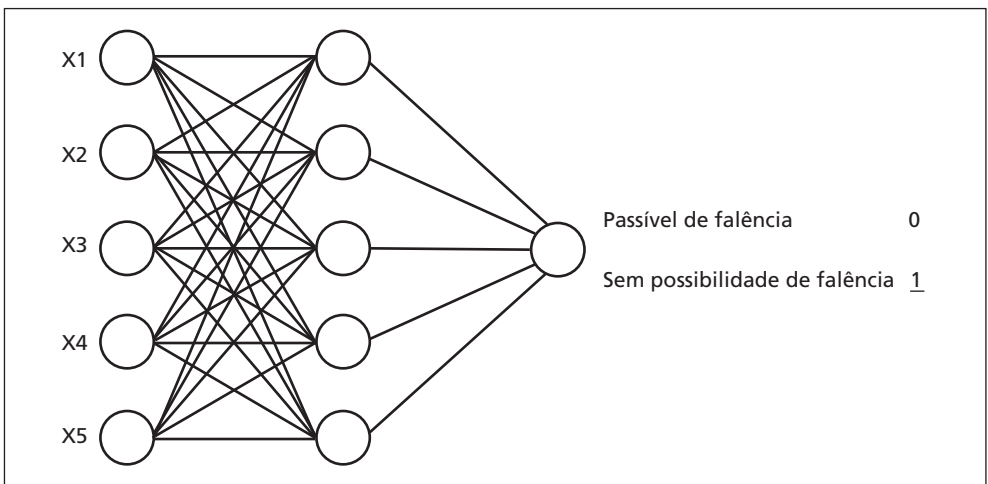


Figura 6.12 Arquitetura da rede neural para predição de falência.

Escolha do algoritmo de aprendizado

Depois que a estrutura da rede é escolhida, precisamos encontrar um algoritmo de aprendizado para identificar um conjunto de pesos de conexão que melhor abranja os dados de treinamento e tenha a melhor precisão preditiva. Para a topologia progressiva escolhida para o problema de previsão de falência, um método típico é usar o algoritmo de retropropagação. Como muitos pacotes comerciais estão disponíveis no mercado, não há necessidade de implementarmos o algoritmo de aprendizado. Pelo contrário, podemos escolher um pacote comercial apropriado para analisar os dados. O quadro *Insights* de tecnologia 6.4 resume informações sobre diferentes tipos de pacotes de software de rede neural disponíveis.

INSIGHTS DE TECNOLOGIA 6.4

Software de RNA

Existem muitas ferramentas para o desenvolvimento de redes neurais (consulte o website deste livro e as listas periódicas de fontes na *PC AI*, pcai.com). Algumas dessas ferramentas funcionam como ambientes de sistema especialista. Elas fornecem um conjunto de arquiteturas, algoritmos de aprendizado e parâmetros padrão, junto com a capacidade de manipular dados. Algumas ferramentas de desenvolvimento podem dar suporte a muitos paradigmas de rede e algoritmos de aprendizado.

Implementações de redes neurais também estão disponíveis na maioria das ferramentas abrangentes de *data mining*, como SAS Enterprise Miner, Clementine e STATISTICA Data Miner. O WEKA é um acervo de algoritmos de aprendizado de máquina com código aberto para tarefas de *data mining*, e inclui recursos de rede neural. É possível fazer o download do WEKA em cs.waikato.ac.nz/~ml/weka. O STATISTICA está disponível em versão de teste para os usuários deste livro.

Muitas ferramentas especializadas de rede neural permitem a criação e implantação de um modelo de rede neural na prática. Qualquer lista com essas ferramentas estaria incompleta. Fontes on-line como a Wikipedia (en.wikipedia.org/wiki/Artificial_neural_network), o diretório de *software* do Google ou do Yahoo, e as listas de fornecedores em pcai.com são bons lugares para localizar informações recentes sobre fornecedores de *software* de rede neural. Alguns dos fornecedores que estavam disponíveis por um tempo ou que comunicaram aplicações industriais dos seus *softwares* de rede neural incluem California Scientific (BrainMaker), NeuralWare, NeuroDimension Inc., Ward Systems Group (Neuroshell) e Megaputer. Novamente, a lista nunca estará completa.

Algumas ferramentas de desenvolvimento de RNA são suplementos de planilha. A maioria consegue ler planilhas, banco de dados e arquivos de texto.

Algumas são *freeware* ou *shareware*. Alguns sistemas de RNA foram desenvolvidos em Java para serem executados diretamente na Web e estão acessíveis através da interface do navegador da Web. Outros produtos de RNA são projetados para interface com sistemas especialistas como produtos híbridos de desenvolvimento.

Em vez disso, os desenvolvedores podem preferir usar linguagens de programação mais comuns, como C++, ou uma planilha para programar o modelo e fazer os cálculos. Uma variação disso seria o uso de uma biblioteca com rotinas de RNA. Por exemplo, a *hav.Software* (hav.com) oferece uma biblioteca de classes C++ para implementação de redes neurais progressivas independentes ou integradas, recorrentes simples e recorrentes de ordem aleatória. Um software computacional como o MATLAB também inclui bibliotecas específicas para redes neurais.

Na prática, como as redes neurais são implementadas? Após um analista/desenvolvedor ter conduzido testes suficientes para certificar que uma rede neural pode fazer um bom trabalho para a aplicação, a rede precisa ser implementada nos sistemas existentes. Uma série de ambientes de rede neural pode produzir códigos, em C++, Java ou Visual Basic, que podem ser integrados a outro sistema que consegue acessar dados-fonte ou é chamado diretamente por uma interface gráfica de usuário para implantação, independentemente do sistema de desenvolvimento. Ou, após treinar uma RNA em uma ferramenta de desenvolvimento, com os pesos, a estrutura de rede e função de transferência, pode-se facilmente desenvolver sua própria implementação em uma linguagem de programação da terceira geração, como a C++. Grande parte dos pacotes de desenvolvimento de RNA, assim como as ferramentas de *data mining*, consegue gerar esses códigos. Então, o código pode ser integrado a uma aplicação autônoma ou em uma aplicação de servidor Web.

Treinamento da rede

O treinamento da RNA é um processo repetitivo que começa a partir de um conjunto aleatório de pesos e gradualmente aprimora a aptidão do modelo da rede e o conjunto de dados conhecido. A iteração continua até o total de erros ser convergido abaixo de um nível aceitável predefinido. No algoritmo de retropropagação, dois parâmetros, *taxa de aprendizado* e *momentum*, podem ser ajustados para controlar a velocidade de obtenção de uma solução. Eles determinam a proporção da diferença entre o valor calculado e o valor real dos casos de treinamento. Alguns pacotes de software podem ter seus próprios parâmetros nas suas heurísticas de aprendizado a fim de acelerar o processo de aprendizagem. É importante ler atentamente antes de usar esse tipo de software.

No processo de treinamento, talvez seja necessário fazer alguma conversão de dados. Isso inclui (1) alterar o formato dos dados para cumprir as exigências do *software*, (2) padronizar a escala de dados para tornar os dados mais semelhantes e (3) remover dados problemáticos. Quando o conjunto de dados de treinamento está pronto, ele é carregado em um pacote e o procedimento de aprendizagem é executado. Dependendo do número de nós e do tamanho do conjunto de dados de treinamento, pode levar de algumas mil até milhões de iterações para obter uma solução.

Teste

Na etapa 2 do processo de desenvolvimento mostrada na Figura 6.11, os dados disponíveis são divididos em conjuntos de dados de treinamento e de teste. Quando o treinamento for completado, será necessário testar a rede. O teste (etapa 8) verifica o desempenho do modelo de rede obtido por meio da avaliação da capacidade de classificar corretamente os dados de teste. O **teste de caixa preta** (ou seja, comparar os resultados do teste com os resultados históricos) é o principal método para verificar que as entradas geram as saídas adequadas. Termos de erro podem ser usados para comparar os resultados em relação aos métodos referenciais conhecidos.

Geralmente, não se supõe que a rede funcione perfeitamente (é difícil de se obter erro zero, se não impossível), e somente um determinado nível de precisão é exigido. Por exemplo, se 1 significa sem possibilidade de falência e 0 significa passível de falência, então qualquer saída entre 0,1 e 1 poderia indicar certa propensão a não falência. A aplicação de rede neural normalmente é uma alternativa para outro método que pode ser usado como referência em relação ao qual comparar a precisão. Por exemplo, uma técnica estatística como regressão múltipla ou outro método quantitativo pode ser conhecido por classificar as entradas corretamente 50% das vezes.

Muitas vezes, a implementação da rede neural aprimora isso. Por exemplo, Liang (1992) relatou que o desempenho da RNA era superior ao desempenho de múltiplas análises discriminantes e indução de regras. Ainscough e Aronson (1999) investigaram a aplicação de modelos de rede neural na predição de vendas varejistas, dado um conjunto com diversas entradas (p.ex., preço regular, promoções variadas). Eles compararam seus resultados com aqueles de regressão múltipla e aprimoraram o R^2 (coeficiente de correlação) ajustado de 0,5 para 0,7. Se a rede neural estiver substituindo as operações manuais, os níveis de desempenho e a velocidade de processamento humano podem ser o padrão para decidir se a fase de teste foi bem-sucedida.

O plano de teste deve incluir casos de rotina bem como situações potencialmente problemáticas. Se o teste revelar grandes desvios, o conjunto de treinamento deve ser reexaminado, e o processo de treinamento talvez tenha de ser repetido (alguns dados “ruins” talvez tenham de ser omitidos do conjunto de entrada).

Observe que podemos equiparar os resultados da rede neural exatamente com aqueles encontrados através de métodos estatísticos. Por exemplo, em regressão linear passo-a-passo, variáveis de entrada algumas vezes são determinadas como insignificantes, mas por causa da natureza da computação neural, uma rede neural as usa para atingir níveis maiores de precisão. Quando elas são omitidas de um modelo de rede neural, seu desempenho normalmente é afetado.

Implementação de uma RNA

Freqüentemente, a implementação de uma RNA (etapa 9) exige interfaces com outros sistemas de informação baseados em computador e treinamento de usuário. Monitoramento e *feedback* permanentes para os desenvolvedores são recomendados de modo a obter melhorias no sistema e êxito a longo prazo. Também é importante ganhar a confiança dos usuários e da gerência no início da implantação para garantir que o sistema seja aceito e usado adequadamente.

Questões de revisão da Seção 6.4

1. Liste as nove etapas na condução de um projeto de rede neural.
2. Quais são alguns dos parâmetros de criação para o desenvolvimento de uma rede neural?
3. Descreva diferentes tipos de software de rede neural disponíveis hoje.
4. Na prática, como as redes neurais são implementadas quando o treinamento/teste é completado?
5. Quais parâmetros talvez precisem ser ajustados no processo de treinamento da rede neural?

6.5

UM EXEMPLO DE PROJETO DE REDE NEURAL

A seguir, descreveremos uma aplicação típica de redes neurais a fim de prever falência de empresas usando os mesmos dados e um esquema experimental semelhante ao usado por Wilson e Sharda (1994). Para fins comparativos, o desempenho das redes neurais será contrastado com o da regressão logística.

O estudo de Altman (1968) tem sido usado como o padrão de comparação para muitos estudos de classificação de falência que usam análise discriminante e regressão logística; estudos de acompanhamento têm identificado outros atributos para melhorar o desempenho da predição. Usamos os mesmos índices financeiros do estudo de Altman, percebendo que entradas mais sofisticadas para o modelo de rede neural devem apenas aprimorar seu desempenho. Os índices são os seguintes:

- X_1 : capital de giro/total de ativos
- X_2 : lucros acumulados/total de ativos
- X_3 : lucros antes de juros e impostos/total de ativos
- X_4 : valor de mercado do patrimônio/dívida total
- X_5 : vendas/total de vendas

A etapa 1 consiste na coleta de dados relevantes. A amostra de empresas para cada um desses índices foi obtida do *Moody's Industrial Manuals*. O manual constitui-se de empresas que ou estavam em funcionamento ou faliram entre 1975 e 1982. A amostra tem um total de 129 empresas, 65 delas faliram durante o período e 64 empresas não falidas coincidiram em setor e ano. Os dados usados para as empresas falidas são das últimas demonstrações financeiras publicadas antes das empresas declararem falência. Portanto, a predição de falência será feita com cerca de 1 ano de antecedência.

A etapa 2 exige que separemos o conjunto de dados em um conjunto de treinamento e um conjunto de teste. Como a determinação da divisão pode afetar descobertas experimentais, um procedimento de reamostragem pode ser usado para criar muitos pares diferentes de conjuntos de treinamento e teste, o que também garante que não há sobreposição na composição dos conjuntos equilibrados de treinamento e teste. Por exemplo, um conjunto de treinamento com 20 padrões pode ser criado aleatoriamente designando 20 registros do conjunto coletado. Um conjunto com outros 20 padrões/registros pode ser criado como um conjunto de teste.

Além disso, os resultados desse (e de qualquer outro) estudo poderiam ser afetados pela proporção de empresas falidas em relação às não falidas tanto no conjunto de treinamento como no conjunto de teste; isto é, o grupo de todas as empresas contém uma determinada proporção de empresas prestes a falir. Esse índice de base pode influenciar no desempenho de uma técnica de

predição de duas maneiras. Primeiro, uma técnica pode não funcionar bem quando as empresas de interesse (ou seja, aquelas que estão falidas) constituem uma porcentagem muito pequena da população (ou seja, baixo índice de base). Isso seria devido a uma incapacidade da técnica de identificar os itens necessários para classificação. Segundo, existem diferenças nos índices de base entre amostras de treinamento e de teste. Se um modelo de classificação for criado usando uma amostra de treinamento com um determinado índice de base, o modelo ainda funcionará quando o índice de base na população de teste for diferente? Essa questão é importante por mais um motivo: se um modelo de classificação baseado em um determinado índice de base funcionar sobre outras proporções, talvez seja possível criar um modelo usando uma proporção maior de casos de interesse do que realmente acontece na população.

Para estudar os efeitos dessa proporção no desempenho preditivo das duas técnicas, criamos três proporções (ou índices de base) para a composição do conjunto de teste enquanto conservamos a composição do conjunto de treinamento fixada em um índice de base de 50/50. O primeiro nível de fator (ou índice de base) pode ser uma proporção de 50/50 de falência para casos de não falência, o segundo nível poderia ser uma proporção de 80/20 (80% de não falência, 20% de falência) e o terceiro nível poderia ser uma proporção aproximada de 90/10. Não sabemos realmente a proporção real de empresas que vão à falência, os casos de 80/20 e 90/10 devem estar perto.

Dentro de cada uma das três diferentes composições do conjunto de teste, 20 diferentes pares do conjunto de teste/treinamento poderiam ser gerados através da reamostragem de Monte Carlo a partir das 129 empresas originais. Portanto, um total de 60 pares distintos do conjunto de dados de treinamento e teste foram gerados a partir dos dados originais. Em cada caso, os pares dos conjuntos de treinamento e de teste continham empresas exclusivas (ou seja, não foi permitida sobreposição). Essa restrição fornece um teste mais seguro do desempenho de uma técnica. Resumindo, redes neurais e modelos de regressão logística são desenvolvidos usando conjuntos de treinamento com proporções iguais de empresas a fim de determinar a função de classificação, mas são avaliadas com conjuntos de teste que contêm índices de base de 50/50, 80/20 e 90/10. (O conjunto de dados usado aqui está disponível no site do livro.)

As etapas 3 a 6 referem-se à preparação para um experimento de rede neural. Podemos usar qualquer pacote de software de rede neural que implemente o algoritmo de aprendizado de retropropagação anteriormente mencionado, de modo a conceber e testar modelos de rede neural treinados. Teríamos de decidir sobre o tamanho da rede neural, incluindo o número de camadas escondidas e de neurônios na camada escondida. Por exemplo, uma estrutura possível de ser usada aqui é: 5 neurônios de entrada (1 para cada índice financeiro), 10 neurônios escondidos e 2 neurônios de saída (1 indicando uma empresa falida e outro, uma não falida). (A Figura 6.13 ilustra essa configuração de rede.) Os valores da saída neural variam entre 0 e 1. O nó de saída BR indica uma empresa a ser classificada como propensa a ir à falência, e o nó NBR, não propensa.

Um usuário de uma rede neural tem duas decisões difíceis para tomar no processo de treinamento (etapa 6): em que ponto a rede neural aprendeu adequadamente as relações, e qual é o limite de erro quanto à determinação de classificações corretas do conjunto de teste? Essas questões são enfrentadas com o uso de tolerâncias de treinamento e de teste que determinam os níveis aceitáveis de variação para considerar as classificações “corretas”.

A etapa 7 refere-se ao treinamento real da rede neural. No treinamento das redes nesse exemplo, um algoritmo heurístico de retropropagação foi usado para garantir a convergência (ou seja, todas as empresas no conjunto de treinamento classificadas corretamente). O conjunto de treinamento foi repetidamente apresentado ao software de rede neural até que o software tivesse aprendido competentemente a relação entre os atributos dos casos e se a empresa estava em perigo. Em seguida, para avaliar com precisão a eficácia da predição da rede, a amostra de teste (ou seja, conjunto de teste) é apresentada à rede, e o número de classificações corretas é registrado (etapa 8).

Na determinação de classificações corretas, um limite de teste de 0,49 foi usado. Dessa maneira, o nó de saída com um valor superior a 0,5 foi usado para avaliar se a rede forneceu uma

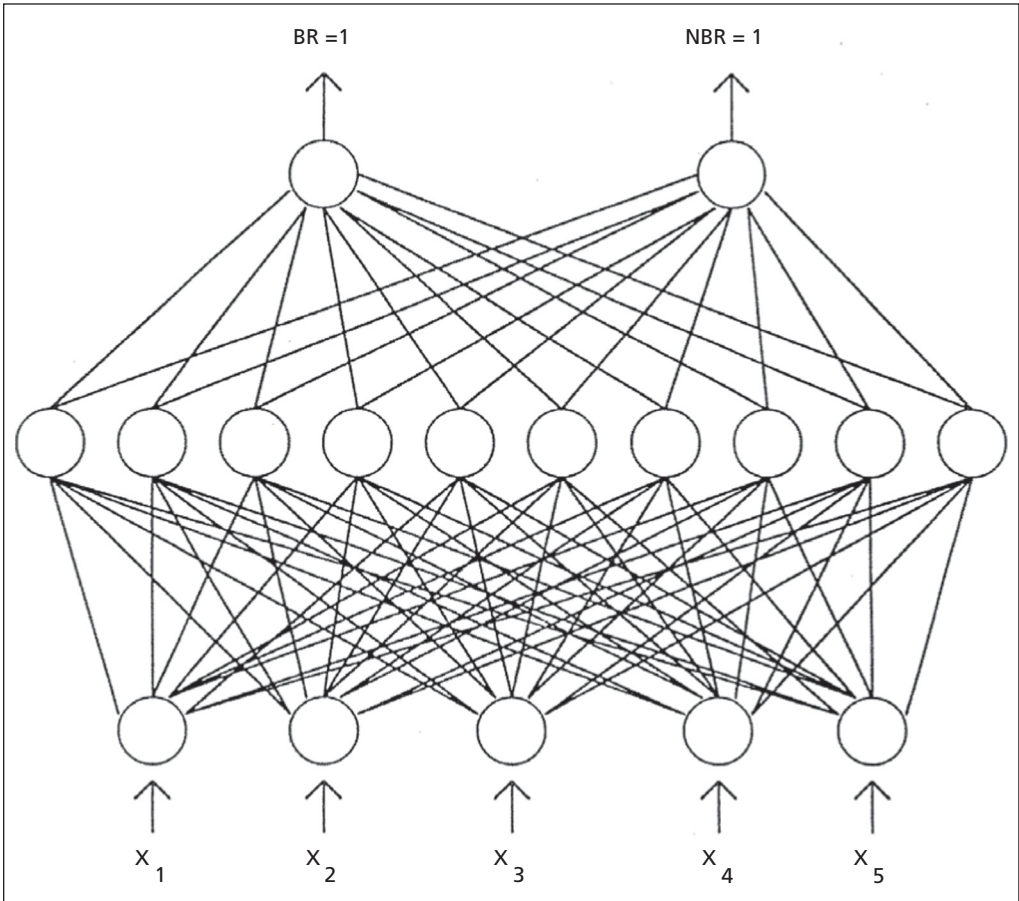


Figura 6.13 Um modelo de rede neural típico para predição de falência.

classificação correta. Casos nos quais ambos os neurônios de saída forneceram níveis de saída ou menores do que 0,5 ou maiores do que 0,5 foram considerados classificações errôneas.

Para comparar o desempenho da rede neural em oposição ao uso de técnicas estatísticas clássicas, um método de regressão logística foi implementado pelo SYSTAT, um pacote de software estatístico. A Tabela 6.2 representa a porcentagem média de classificações corretas fornecidas pelas duas diferentes técnicas quando avaliadas pelas 20 amostras de teste para cada um dos três diferentes índices de base do conjunto de teste. Quando os conjuntos de teste continham um número igual dos dois casos, as redes neurais classificaram corretamente 97,5% dos casos de teste, enquanto que a regressão logística esteve correta 93,25% das vezes. De maneira semelhante, quando os conjuntos de teste tinham 20.070 empresas falidas, as redes neurais classificaram com um índice de 95,6% de acerto, enquanto que a regressão logística classificou corretamente com um índice de 92,2%.

Um teste não-paramétrico, o teste de Wilcoxon para observações correlacionadas, foi realizado para avaliar se as porcentagens de classificação correta para as duas técnicas eram significativamente diferentes. Aquelas ocorrências nas quais diferenças estatisticamente significativas foram encontradas são indicadas pelas notas de rodapé na Tabela 6.2. Em geral, as redes neurais tiveram um desempenho significativamente melhor do que a regressão logística.

A Tabela 6.2 também ilustra as porcentagens corretas de predições de empresas falidas e não falidas. Na predição dos casos de falência, as redes neurais fizeram predições significativamente

Tabela 6.2 Comparação de desempenho entre redes neurais e regressão logística

Critérios	Proporções de teste					
	50/50		80/20		90/10	
	RN	RL	RN	RL	RN	RL
Porcentagem total de classificação correta	97,5 ^a	93,25	95,6 ^a	92,2	95,68 ^b	90,23
Índice de êxito na classificação de empresas passíveis de falência	97,0 ^a	91,90	92,0	92,0	92,5	95,0 ($p = 0,282$)
Índice de êxito na classificação de empresas sem possibilidade de falência	98,0 ^a	95,5	96,5 ^a	92,25	96,0 ^b	89,75

^a $p < 0,01$.

^b $p < 0,05$.

melhores do que a regressão logística para conjuntos de teste de proporção igual, na mesma porcentagem quando o índice era 80/20, e um pouco pior (embora não significativamente) para conjuntos de teste de 90/10. As redes neurais nitidamente superaram em desempenho o modelo de regressão logística quanto à predição de empresas não passíveis de falência.

Uma série de estudos no passado recente investigou o desempenho de redes neurais na predição de fracasso empresarial. Esses estudos compararam o desempenho da rede neural ao desempenho das técnicas estatísticas tradicionais, como análise discriminante e regressão logística. Além disso, alguns estudos compararam redes neurais a outras técnicas de inteligência artificial, como métodos de aprendizado indutivo (p.ex., ID3). O objetivo desta seção é ilustrar como um projeto de rede neural pode ser concluído, não necessariamente argumentar que as redes neurais saem-se melhor nesse domínio de problema.

Questões de revisão da Seção 6.5

1. Quais parâmetros podem ser usados para prever a falência de uma empresa?
2. Como os dados foram divididos entre conjuntos de treinamento e de teste para esse experimento?
3. Explique o que significa *reamostragem* neste contexto. Como a reamostragem foi usada para esse problema?
4. Quais foram os parâmetros de rede para esse experimento de rede neural?
5. Como uma saída foi convertida para significar falência ou não falência?
6. Como o modelo de rede neural se compara a um modelo de regressão logística nesse experimento?

6.6

OUTROS PARADIGMAS DAS REDES NEURAIS

As redes neurais baseadas em MLP descritas até aqui neste capítulo são apenas um tipo específico de redes neurais. Literalmente centenas de redes neurais diferentes foram propostas. Muitas são variantes do modelo MLP já visto; elas diferem-se apenas nas suas implementações de representação de entrada, processo de aprendizagem, processamento de saída e assim por diante. Mas existem muitos tipos de redes neurais diferentes do modelo MLP. Alguns deles serão apresentados ainda neste capítulo. Outros incluem redes com função de base radial, redes neurais probabilísticas, redes neurais de regressão generalizada e máquinas de vetores de suporte. Diversos recursos

on-line descrevem detalhes dos tipos de redes neurais. Uma boa fonte apresentada no Capítulo 4 é o e-book da StatSoft (statsoft.com/textbook/stathome.html). A próxima subseção introduz algumas das variedades clássicas de redes neurais.

Redes de Hopfield

Um modelo de rede neural interessante é a rede de Hopfield (Hopfield, 1982). John Hopfield mostrou em uma série de artigos nos anos 1980 como redes altamente interconectadas ou neurônios não-lineares podem ser extremamente eficazes na computação. Essas redes forneciam uma rápida solução calculada para problemas declarados em termos de condições ideais desejadas, muitas vezes sujeitas a restrições.

Uma rede de Hopfield comum é uma única grande camada de neurônios com total interconectividade – isto é, cada neurônio está conectado a um neurônio sim e outro não. Além disso, a saída de cada neurônio pode depender dos seus valores anteriores. Um uso da rede de Hopfield tem sido na resolução de problemas de otimização limitada, como o clássico problema do caixeiro viajante (TSP). Nesse tipo de aplicação, cada neurônio representa a conveniência de uma cidade n a ser visitada na posição m de um roteiro do TSP. Os pesos de interconexão estão especificados, representando as limitações da solução viável para o TSP (p.ex., forçar uma cidade a aparecer em um roteiro somente uma vez). Uma função de energia está especificada, a qual representa o objetivo do processo de solução do modelo (p.ex., minimizar a distância total no roteiro do TSP) e é usada na determinação de quando parar a evolução da rede neural em um estado final. A rede começa com valores aleatórios de neurônio e, usando os pesos de interconexão definidos, os valores de neurônio são atualizados ao longo do tempo. Gradualmente, os valores de neurônio estabilizam-se, evoluindo para um estado final (conforme orientado pela função geral de energia) que representa uma solução para o problema. Nesse ponto de evolução da rede, o valor de neurônio (n, m) representa se a cidade n deveria ser na localização m do roteiro do TSP. Enquanto Hopfield e Tank (1985) e outros alegavam grande êxito na resolução do TSP, pesquisas adicionais mostraram que essas afirmações eram um tanto precipitadas. Todavia, essa abordagem nova para um problema clássico tem potencial para os problemas de otimização, especialmente quando a tecnologia possibilita tirar vantagem do paralelismo inerente das redes neurais.

As redes de Hopfield são diferentes das redes progressivas porque os neurônios estão altamente interconectados, os pesos entre os neurônios tendem a ser fixos, e não há treinamento propriamente dito. A complexidade e o desafio de usar uma rede de Hopfield para problemas de otimização estão na especificação correta dos pesos de interconexão e na identificação da função geral de energia apropriada para orientar o processo de evolução da rede.

Redes auto-organizáveis

Também conhecida como rede auto-organizável, a rede de Kohonen é outro modelo de rede neural. Essas redes aprendem de maneira não supervisionada. A base biológica desses modelos é a hipótese de que alguma organização acontece no cérebro humano quando um estímulo externo é fornecido. O algoritmo de Kohonen forma “mapas de características”, onde regiões de neurônios são construídas. Essas regiões são organizadas de tal maneira que neurônios topologicamente próximos são sensíveis às entradas semelhantes dentro do modelo. Mapas auto-organizáveis, ou mapas de características auto-organizáveis, às vezes podem ser usados para desenvolver alguma compreensão inicial dentro dos dados. Por exemplo, mapas auto-organizáveis poderiam aprender a identificar grupos de dados de modo que um analista pudesse criar modelos mais sofisticados para cada subgrupo/grupo. Em casos nos quais o analista não tem uma boa idéia do número de classes ou saída ou classe de saída real para qualquer padrão dado, os mapas auto-organizáveis podem funcionar.

Questões de revisão da Seção 6.6

1. Liste alguns dos diferentes tipos de redes neurais.
2. Qual é uma diferença importante entre uma rede MLP e uma rede de Kohonen?
3. Qual é o outro nome para a rede de Kohonen?
4. Faça uma breve descrição da rede de Hopfield.

6.7

APLICAÇÕES DE RNA

A RNA tem sido aplicada em muitos campos. Um levantamento das suas aplicações em finanças pode ser encontrado em Fadlalla e Lin (2001). Foram feitos diversos testes de redes neurais em mercados financeiros. Collard (1990) afirmou que seu modelo de rede neural para treinamento com mercadorias teria resultado em lucros significativos sobre outras estratégias de negociação. Kamijo e Tanigawa (1990) usaram uma rede neural para fazer um gráfico com os dados da Bolsa de Valores de Tóquio. Eles descobriram que os resultados do modelo superariam uma estratégia de “comprar e manter”. Por último, foi desenvolvido um modelo neural para predição de mudança da porcentagem no S&P 500 com cinco dias de antecedência, usando uma variedade de indicadores econômicos (Fishman et al., 1991). Os autores afirmam que o modelo tem fornecido predições mais precisas do que alegaram especialistas na área usando os mesmos indicadores.

Redes neurais têm sido treinadas com êxito para determinar se solicitações de empréstimo devem ser aprovadas (Gallant, 1988). Também tem sido mostrado que as redes neurais podem prever a solvência de candidatos a financiamento da casa própria melhor do que os agentes hipotecários (Collins et al., 1988). Prever a classificação das obrigações ao portador e tentar prever sua rentabilidade é outra área onde as redes neurais têm sido aplicadas com êxito (consulte Dutta e Shakhar, 1988; e Surkan e Singleton, 1990). As redes neurais superaram em desempenho a análise de regressão e outras ferramentas de modelagem matemática na predição da classificação e rentabilidade das obrigações. A principal conclusão obtida foi que as redes neurais forneceram uma estrutura mais geral para vincular a informação financeira de uma empresa com a classificação da obrigação correspondente.

Prevenção contra fraude é outra área de aplicação da rede neural nos negócios. O Chase Manhattan Bank usou de maneira bem-sucedida redes neurais para lidar com fraudes de cartão de crédito (Rochester, 1990), com os modelos de rede neural superando tradicionais abordagens de regressão. Além disso, as redes neurais têm sido usadas na validação de assinaturas em bancos (consulte Francett, 1989; e Mighell, 1989). Essas redes identificaram falsificações significativamente melhor do que qualquer perito humano.

Outra área importante de aplicação estatística das redes neurais é na previsão de séries temporais. Diversos estudos tentaram usar redes neurais para predição de séries temporais. Exemplos incluem Fozzard et al. (1989), Tang et al. (1991) e Hill et al. (1994). A conclusão geral é que as redes neurais parecem funcionar, pelo menos, tão bem quanto a técnica de previsão de Box-Jenkins.

Como as redes neurais têm sido um assunto de estudo profundo desde o final dos anos 80, muitas aplicações e também experimentos com aplicações foram desenvolvidos. Pode-se fazer pesquisas simples na Internet para encontrar exemplos recentes em acréscimo àqueles listados neste capítulo. Outros relatórios recentes incluem rastreamento ativo de invasão (consulte Thaler, 2002), filtragem de conteúdo da Web (Lee et al., 2002), predição da taxa de câmbio (Davis et al., 2001), e alocação de leitos em hospital (Walczak et al., 2002). Aplicações mais novas estão surgindo nas áreas da medicina e saúde. Veja o Caso de Aplicação 6.5 para obter um exemplo.

Em geral, as RNAs são adequadas para problemas cujas entradas são tanto categóricas quanto numéricas, e onde as relações entre entradas e saídas não são lineares ou os dados de entrada não

CASO DE APLICAÇÃO 6.5**Redes neurais para diagnóstico de câncer de mama**

A RNA demonstrou ser uma ferramenta útil nas tarefas de classificação e de reconhecimento de padrões em várias áreas, incluindo medicina clínica. Apesar da vasta aplicabilidade da RNA, a grande quantidade de dados exigidos para treinamento torna o uso delas uma técnica de classificação inadequada quando os dados disponíveis são escassos. A espectroscopia por ressonância magnética (ERM) desempenha um papel fundamental na investigação da bioquímica da célula e oferece um método confiável para detecção de mudanças metabólicas no tecido mamário. A escassez de dados e a complexidade de interpretação de informações fisiológicas relevantes impõem exigências adicionais que impedem a aplicabilidade da maioria das técnicas estatísticas e de aprendizado de máquina desenvolvidas. Redes neurais artificiais baseadas em conhecimento (KBANN) ajudam a superar essas dificuldades e complexidades. Uma KBANN combina conhecimento de uma área, em forma de regras simples, com aprendizado conexionista. Essa combinação treina a rede por meio do uso de pequenos conjuntos de dados (como é característico das tarefas de diagnóstico médico). A estrutura primária é baseada nas dependências de um conjunto de regras conhecidas da área, e é necessário refinar essas regras através do treinamento.

O processo da KBANN consiste de dois algoritmos. Um é o algoritmo Regras-para-rede, no qual a tarefa principal é o processo de tradução entre uma base de conhecimento que contém informações sobre uma teoria de domínios e a estrutura inicial de uma

rede neural. Esse algoritmo mapeia a estrutura de uma teoria dos domínios aproximadamente correta, com todas as regras e suas dependências, dentro de uma rede neural. Então, a rede definida é treinada usando o algoritmo de aprendizado de retropropagação.

Mecanismos de regeneração, que inibem ou estimulam o crescimento de células normais, controlam a divisão e reposição de células em tecidos normais. No caso dos tumores, esse processo é incapaz de controlar a produção de células novas, e a divisão é feita sem qualquer consideração à necessidade de reposição, atrapalhando a estrutura do tecido normal. Alterações observadas nas concentrações do metabólito de fosfolípide, as quais estão associadas a diferenças na proliferação celular em tecidos malignos, serviram como entradas básicas para a identificação de aspectos relevantes presentes em tecidos malignos ou cancerosos, mas não em tecidos normais. Os níveis anormais de determinadas características do fosfolípide são considerados indicadores de tumor. Eles incluem diversos parâmetros, como PDE, PME, Pi, PCr, γ ATP, α ATP e β ATP. A KBANN produziu uma classificação precisa de tumor de 87% a partir de um conjunto de 26, com uma média de erro padrão de 0,0500 e um desvio padrão de 0,0179.

Fontes: M. Sordo, H. Buxton e D. Watson, "A Hybrid Approach to Breast Cancer Diagnosis", em *Practical Applications of Computational Intelligence Techniques*, Vol. 16, em L. Jain e P. DeWilde (eds.), Kluwer, Norwell, MA, 2001, acl.icnet.uk/PUBLICATIONS/sordo/chapter2001.pdf (acessado em março de 2006).

estão regularmente distribuídos. Nesses casos, métodos estatísticos clássicos podem não ser suficientemente confiáveis. Como a RNA não faz quaisquer suposições sobre a distribuição de dados, sua capacidade é menos afetada do que nos métodos estatísticos tradicionais quando os dados não estão adequadamente distribuídos. Por último, existem casos nos quais as redes neurais simplesmente oferecem mais uma maneira de criar um modelo preditivo para a situação em questão. Dada a facilidade de experimentação usando as ferramentas de software disponíveis, é útil investigar a capacidade das redes neurais em qualquer situação de modelagem de dados.

Questões de revisão da Seção 6.7

1. Liste algumas aplicações de rede neurais em contabilidade/finanças.
2. Quais são algumas aplicações de redes neurais em engenharia?
3. Como as redes neurais têm sido usadas na área da saúde?
4. Quais são algumas aplicações de redes neurais em segurança da informação?
5. Faça uma pesquisa na Web para identificar aplicações de redes neurais na segurança nacional.

6.8

DEMONSTRAÇÃO DE UM SOFTWARE DE REDE NEURAL

O Tutorial online T4 oferece uma demonstração de uso do *software* de redes neurais. Essa seção é usada, com permissão, a partir do tutorial do *software* STATISTICA. Alunos e professores que usam este livro estão aptos a receber uma licença de seis meses a fim de usar o *software* STATISTICA para concluir os exercícios dos Capítulos 4 e 6. A solicitação dessa cópia do *software* deve ser feita pelo instrutor por meio do preenchimento do cupom disponível no *site* parceiro www.pre-nhall.com/turban. Observe que projetos semelhantes de *software* também podem ser concluídos com o uso de ferramentas identificadas no quadro *Insights* de tecnologia 6.4.

Destaques do capítulo

- Computação neural envolve um conjunto de métodos que emulam a maneira como o cérebro humano funciona. A unidade básica de processamento é o neurônio. Múltiplos neurônios estão ligados e agrupados dentro de camadas.
- Em uma rede neural, o conhecimento está armazenado no peso associado a cada conexão entre dois neurônios.
- Retropropagação é o paradigma mais popular nas aplicações comerciais de redes neurais. A maioria das aplicações comerciais é controlada usando esse algoritmo.
- Uma rede neural baseada em retropropagação é composta de uma camada de entrada, uma camada de saída e um determinado número de camadas escondidas (geralmente uma). Os nós em uma camada estão totalmente conectados aos nós da próxima camada. O aprendizado é feito através de um processo de tentativa e erro para ajuste dos pesos de conexão.
- Cada nó na camada de entrada representa um único atributo que pode afetar a predição.
- O aprendizado da rede neural pode ocorrer de modo supervisionado ou não supervisionado.
- No modo de aprendizado supervisionado, os padrões de treinamento incluem uma resposta correta/classificação/previsão.
- No modo de aprendizado não supervisionado, não há respostas. Por isso, o aprendizado não supervisionado é usado para agrupamento ou análise exploratória dos dados.
- O processo comum de aprendizagem em uma rede neural envolve três etapas: (1) calcular saídas temporárias com base nas entradas e nos pesos aleatórios, (2) calcular saídas com metas desejadas e (3) ajustar os pesos e repetir o processo.
- A regra delta normalmente é usada para ajustar os pesos. Inclui uma taxa de aprendizado e um parâmetro de *momentum*.
- Desenvolver sistemas baseados em rede neural exige um processo passo a passo. O processo inclui preparação e pré-processamento dos dados, treinamento e teste, e conversão do modelo treinado em um sistema de produção.
- O software de rede neural está disponível para possibilitar uma experiência acessível com muitos modelos. Módulos de rede neural estão incluídos em todas as principais ferramentas de software de *data mining*. Pacotes específicos de rede neural também estão disponíveis. Algumas ferramentas de rede neural estão disponíveis como suplementos de planilha.
- Após uma rede treinada ter sido criada, geralmente é implementada em sistemas para usuários finais por meio de linguagens de programação, como C++, Java e Visual Basic. A maioria das ferramentas de rede neural consegue gerar códigos para a rede treinada nessas linguagens.
- Fora a retropropagação, existem muitos modelos de rede neural, incluindo funções de base radial, máquinas de vetores de suporte, redes de Hopfield e mapas auto-organizáveis de Kohonen.
- Aplicações de rede neural são usadas em quase todas as disciplinas de negócios, bem como em praticamente todas as outras áreas funcionais.
- Aplicações comerciais de rede neural incluem finanças, predição de fracasso empresarial, previsão de séries temporais etc.
- Novas aplicações de redes neurais estão surgindo nas áreas da saúde, segurança, etc.

Termos-chave

- | | | |
|----------------------------------|--------------------|----------------------------------|
| • algoritmo de aprendizado | • auto-organização | • computação neural |
| • aprendizado não supervisionado | • axônio | • dendrito |
| • aprendizado supervisionado | • camada escondida | • elemento de processamento (PE) |

- | | | |
|--|--------------------------------|---|
| • função de soma | • núcleo | • sinapse |
| • função de transformação (trans-
ferência) | • perceptron | • taxa de aprendizado |
| • função sigmóide (ativação lógica) | • peso de conexão | • teoria da ressonância adaptativa
(ART) |
| • mapa de características auto-or-
ganizável de Kohonen | • processamento paralelo | • teste de caixa preta |
| • <i>momentum</i> | • reconhecimento de padrão | • topologia |
| • neurônio | • rede neural | • valor limite |
| | • rede neural artificial (RNA) | |
| | • retropropagação | |

QUESTÕES PARA DISCUSSÃO

1. Compare a rede neural biológica com a rede neural artificial. Quais aspectos das redes biológicas não são imitados pelas redes artificiais? Quais aspectos são semelhantes?
2. O desempenho da RNA depende das funções de soma e transformação. Explique os efeitos combinados das funções de soma e transformação e como elas diferem da análise de regressão estatística.
3. A RNA pode ser usada para aprendizado supervisionado e não supervisionado. Explique como elas aprendem no modo supervisionado e no modo não supervisionado.
4. Explique a diferença entre conjunto de treinamento e conjunto de teste. O que precisamos para diferenciá-los? O mesmo conjunto pode ser usado para ambos os fins? Por que ou por que não?
5. Digamos que uma rede neural foi criada para prever a capacidade financeira dos requerentes. Existem dois tipos de nós de saída: um para sim (1 = sim; 0 = não) e um para não (1 = não; 0 = sim). Um requerente recebe uma pontuação de 0,83 para o nó de saída “sim” e 0,44 para o nó de saída “não”. Discuta o que pode ter acontecido e se o requerente é um bom risco de crédito.
6. Todos gostariam de ganhar muito dinheiro no mercado de ações, mas somente alguns são bem-sucedidos. Por que usar uma RNA é um método promissor? O que ela consegue fazer que outras tecnologias de suporte à decisão não conseguem? Como ela poderia falhar?

Exercícios

Exercícios da Teradata University e outros exercícios interativos

1. Acesse o *website* da Teradata Student Network (em teradatastudentnetwork.com) ou a URL fornecida pelo seu instrutor. Localize os seminários na web relacionados a *data mining* e redes neurais. Veja especificamente o seminário dado pelo Professor Hugh Watson na conferência SPIRIT2005 na Universidade do Estado de Oklahoma. Em seguida, responda as seguintes questões:
 - a. Que aplicação em tempo real da Continental Airlines pode ter usado uma rede neural?
 - b. Quais entradas e saídas podem ser usadas na criação de uma aplicação de rede neural?
 - c. Dado que as aplicações de *data mining* da Continental são em tempo real, como a empresa deveria implementar uma rede neural na prática?
 - d. Quais outras aplicações de rede neural você proporia para o setor aéreo?
2. Acesse o *website* da Teradata Student Network (em teradatastudentnetwork.com) ou a URL fornecida pelo seu instrutor. Localize o caso da Harrah's. Leia-o e responda as seguintes questões:
 - a. Qual das aplicações de dados da Harrah's são mais propensas a serem implementadas usando redes neurais?
 - b. Qual outra aplicação a Harrah's poderia desenvolver usando os dados coletados de seus clientes?
 - c. Quais são alguns dos cuidados que você deveria ter como cliente desse cassino?
3. Esse exercício refere-se ao projeto exemplo deste capítulo. O problema de predição de falência pode ser visto como um problema de classificação. O conjunto de dados que você usará nesse problema inclui cinco índices que foram calculados a partir das demonstrações financeiras de empresas reais. Esses cinco índices foram usados em estudos que envolvem predição de falência. A primeira amostra inclui dados sobre empresas que faliram e empresas que não faliram. Essa será sua amostra de treinamento para a rede neural. A segunda amostra com 10 empresas também possui algumas empresas falidas e outras não. Seu objetivo é treinar uma rede

neural, usando os 20 primeiros dados, e então testar o desempenho com os outros 10 dados. (Tente analisar os novos casos você mesmo, manualmente, antes de executar a rede neural e ver quão bem você

trabalha.) As tabelas a seguir mostram a amostra de treinamento e os dados de teste que devem ser usados nesse exercício:

Amostra de treinamento						
Empresa	CG/TA	LA/TA	LAJIR/TA	VMP/DT	V/TV	BR/NB
1	0,165	0,1192	0,2035	0,813	1,6702	1
2	0,1415	0,3868	0,0681	0,5755	1,0579	1
3	0,5804	0,3331	0,081	1,1964	1,3572	1
4	0,2304	0,296	0,1225	0,4102	3,0809	1
5	0,3684	0,3913	0,0524	0,1658	1,1533	1
6	0,1527	0,3344	0,0783	0,7736	1,5046	1
7	0,1126	0,3071	0,0839	1,3429	1,5736	1
8	0,0141	0,2366	0,0905	0,5863	1,4651	1
9	0,222	0,1797	0,1526	0,3459	1,7237	1
10	0,2776	0,2567	0,1642	0,2968	1,8904	1
11	0,2689	0,1729	0,0287	0,1224	0,9277	0
12	0,2039	-0,0476	0,1263	0,8965	1,0457	0
13	0,5056	-0,1951	0,2026	0,538	1,9514	0
14	0,1759	0,1343	0,0946	0,1955	1,9218	0
15	0,3579	0,1515	0,0812	0,1991	1,4582	0
16	0,2845	0,2038	0,0171	0,3357	1,3258	0
17	0,1209	0,2823	-0,0113	0,3157	2,3219	0
18	0,1254	0,1956	0,0079	0,2073	1,489	0
19	0,1777	0,0891	0,0695	0,1924	1,6871	0
20	0,2409	0,166	0,0746	0,2516	1,8524	0

Dados de teste						
Empresa	CG/TA	LA/TA	LAJIR/TA	VMP/DT	V/TV	BR/NB
A	0,1759	0,1343	0,0946	0,1955	1,9218	?
B	0,3732	0,3483	-0,0013	0,3483	1,8223	?
C	0,1725	0,3238	0,104	0,8847	0,5576	?
D	0,163	0,3555	0,011	0,373	2,8307	?
E	0,1904	0,2011	0,1329	0,558	1,6623	?
F	0,1123	0,2288	0,01	0,1884	2,7186	?
G	0,0732	0,3526	0,0587	0,2349	1,7432	?
H	0,2653	0,2683	0,0235	0,5118	1,835	?
I	0,107	0,0787	0,0433	0,1083	1,2051	?
J	0,2921	0,239	0,0673	0,3402	0,9277	?

Descreva os resultados da predição da rede neural, incluindo *software*, arquitetura e informações de treinamento. Envie o(s) arquivo(s) da rede treinada para que seu instrutor possa carregá-la e testá-la.

4. Para esse exercício, seu objetivo é criar um modelo para identificar entradas ou indicadores que diferencie clientes de risco dos outros (baseado nos padrões relativos aos clientes anteriores) e, em seguida, use essas entradas para prever os novos clientes de risco.

Os dados amostrais a serem usados nesse exercício são fornecidos no site www.prenhall.com/turban, nome do arquivo: CreditRisk.xls. O conjunto de dados tem 425 casos e 15 variáveis relativas aos clientes antigos e atuais que, por diversos motivos, fizeram um empréstimo no banco. O conjunto de dados contém várias informações relacionadas ao cliente, motivo para o empréstimo, tempo de emprego, solvência e informações demográficas etc, e por último o resultado ou variável dependente para solvência, classificando cada caso como bom ou ruim, com base na experiência anterior da instituição.

Você deve pegar 400 casos como casos de treinamento e usar os outros 25 para teste. Em seguida, crie um modelo de rede neural para aprender as características do problema e teste o desempenho com os outros 25 casos. Informe sobre o desempenho de aprendizado e teste do seu modelo. Prepare um relatório que identifique a arquitetura da rede neural, os parâmetros de treinamento e o desempenho resultante no conjunto de teste.

(Esse exercício é cortesia da StatSoft, Inc., com base no conjunto de dados em alemão do site [ftp://ftp.ics.uci.edu/pub/machine-learning-databases/statlog/german](http://ftp.ics.uci.edu/pub/machine-learning-databases/statlog/german) alterado e renomeado CreditRisk.)

- Prever a receita de bilheteria para uma determinada produção cinematográfica é um desafio interessante. Apesar da dificuldade associada à natureza imprevisível do domínio de problema, diversos pesquisadores tentaram prever a receita total da bilheteria após o lançamento inicial de um filme no cinema. Nesse problema, você examina a previsão do desempenho financeiro de um filme na bilheteria antes do lançamento no cinema por meio da conversão do problema de previsão em um problema de classificação. Isto é, ao invés de prever a estimativa de auge das receitas de bilheteria, você classifica um filme baseado na sua receita de bilheteria em uma das nove categorias, indo de fracasso a sucesso, levando em conta uma série de fatores decididos pela opinião recebida de especialista do ramo e por estudos anteriores. A lista a seguir mostra as variáveis usadas:

Atributos	Varição de valores	Tipo
Classificação da Associação da indústria cinematográfica (MPAA)	5 categorias de classificação possíveis: G, PG, PG-13, R, NR	Binário (0,1)
Competição	3 pseudo variáveis: alta, média, baixa competição	Binário (0,1)

Valor da estrela	3 variáveis do grau de valor da estrela: A+/A (alto), B (médio), C (insignificante)	Binário (0,1)
Categoria do conteúdo (gênero)	10 categorias: ficção científica, drama épico histórico, drama moderno, drama político, suspense, terror, comédia, desenho, ação, documentário	Binário (0,1)
Efeitos técnicos	3 variáveis binárias independentes: avaliações de efeito técnico alto, médio, baixo	Binário (0,1)
Seqüência	1 variável para especificar se um filme é uma seqüência	Binário (0,1)
Número de cenas	Variável contínua	Número inteiro positivo

Cada variável categórica independente (exceto a variável gênero) é convertida em uma representação binária 1 de N . Por exemplo, as 5 classificações da MPAA são representadas como cinco variáveis 0-1. No processo de atribuição de valor, todas essas pseudo-representações de uma variável categórica recebem o valor de 0, exceto aquela que é verdadeira para o caso atual, para a qual é dado o valor de 1. Para um filme de classificação PG, a segunda variável de entrada está no nível 1, as outras (1 e 3-5) estão no nível 0.

Aqui, a variável de interesse é a receita bruta da bilheteria. Um filme baseado na sua receita de bilheteria está classificado em uma das nove categorias, indo de fracasso a sucesso. A variável dependente pode ser convertida em nove classes usando os seguintes pontos de corte:

Número de classe	Varição (em milhões)
1	< 1 (fracasso)
2	> 1 e < 10
3	> 10 e < 20
4	> 20 e < 40
5	> 40 e < 65
6	> 65 e < 100
7	> 100 e < 150
8	> 150 e < 200
9	> 200 (sucesso)

Faça o *download* dos dados do conjunto de treinamento em www.prenhall.com/turban, nome do arquivo: movietrain.xls, que tem 184 registros e

está em formato Microsoft Excel. Use a descrição dos dados para entender o domínio e o problema que você está tentando resolver. Escolha cuidadosamente suas variáveis independentes; desenvolva pelo menos três modelos de classificação (p.ex., árvore de decisão, regressão logística, redes neurais). Compare a precisão dos resultados (usando técnicas de validação cruzada de 10 níveis e de divisão de porcentagem), use matrizes de confusão e comente o resultado. Teste os modelos desenvolvidos no conjunto de teste (acesse www.prenhall.com/turban, nome do arquivo: movietest.xls, 29 registros) e analise os resultados com modelos diferentes e apresente o melhor modelo de classificação, respaldando-o com os seus resultados.

Tarefas em grupo e interpretação de papéis

1. Considere o seguinte conjunto de dados que se refere ao uso diário de eletricidade como uma função de alta temperatura externa (para o dia):

Temperatura, X	Quilowatts, Y
8,22 °C	12.530
11,16 °C	10.800
12,83 °C	10.180
15,11 °C	9.730
16,61 °C	9.750
19 °C	10.230
21,05 °C	11.160
24,89 °C	13.910
26,27 °C	15.690
26,5 °C	15.110
26,77 °C	17.020
28,5 °C	17.880

- a. Faça um gráfico com os dados brutos. Que padrão você vê? O que você acha que realmente está afetando o uso de eletricidade?
- b. Resolva esse problema com a regressão linear $Y = a + bX$ (em uma planilha). Quão bem ela funciona? Faça um gráfico com os resultados. O que está errado? Calcule o erro da soma dos quadrados e R^2 .
- c. Resolva esse problema usando uma regressão não-linear. Recomendamos uma função quadrática, $Y = a + b_1X + b_2X^2$. Quão bem ela funciona? Faça um gráfico com os resultados. Há alguma coisa errada? Calcule o erro da soma dos quadrados e R^2 .
- d. Divida o problema em três seções (observe o gráfico) e resolva-o usando três modelos de regressão linear – um para cada seção. Quão bem

ele funciona? Faça um gráfico com os resultados. Calcule o erro da soma dos quadrados e R^2 . Esse método de modelagem é adequado? Por que ou por que não?

- e. Crie uma rede neural para resolver o problema original. (Você pode ter de dimensionar os valores de X e Y para estarem entre 0 e 1.) Treine-a (com todo o conjunto de dados) e resolva o problema (ou seja, faça previsões para cada um dos itens originais dos dados). Quão bem ela funciona? Faça um gráfico com os resultados. Calcule o erro da soma dos quadrados e R^2 .
 - f. Qual método funcionou melhor e por quê?
2. Crie uma rede neural real. Usando o *software* baixado da Web (p.ex., Braincel, em promland.com, ou outro site), identifique dados reais (p.ex., comece a procurar na Web em ics.uci.edu/~mllearn/MLRepository.html ou use os dados de uma empresa com a qual alguém do seu grupo possui um contato) e crie uma rede neural para fazer previsões. Os tópicos poderiam incluir previsões de vendas, previsão de sucesso em um programa acadêmico (p.ex., prever a média geral das notas de uma classificação de ensino médio e pontuações do SAT; tenha cuidado com os dados “ruins”, como média geral de 0,0), ou preços de imóveis; ou faça um levantamento na turma para saber peso, gênero e altura a fim de tentar prever a altura com base nos outros dois fatores. (Dica: use os dados do censo (EUA), no *site* deste livro ou em census.gov, por estado, para identificar uma relação entre nível de escolaridade e renda.) Quão boas foram as suas previsões? Compare os resultados com as previsões geradas usando métodos estatísticos padrão (regressão). Qual é o melhor método? Como seu sistema poderia ser integrado a um DSS para tomada de decisão real?
 3. Para cada uma das seguintes aplicações, seria melhor usar uma rede neural ou um sistema especialista? Explique suas respostas, incluindo possíveis exceções ou condições especiais.
 - a. Diagnóstico de uma doença bem-estabelecida porém complexa
 - b. Subsistema PLU (Price-lookup) para um vendedor de grandes volumes de mercadorias
 - c. Sistema de processamento de consulta automática de voz
 - d. Treinamento de funcionários novos
 - e. Reconhecimento de caligrafia
 4. Considere o seguinte conjunto de dados, que inclui três atributos e uma classificação para decisões de admissão em um programa de MBA:

GMAT	Média geral das notas	Percentil quant. do GMAT	Decisão
650	2,75	35	NÃO
580	3,50	70	NÃO
600	3,50	75	SIM
450	2,95	80	NÃO
700	3,25	90	SIM
590	3,50	80	SIM
400	3,85	45	NÃO
640	3,50	75	SIM
540	3,00	60	?
690	2,85	80	?
490	4,00	65	?

- Usando os dados fornecidos aqui como exemplos, desenvolva seu próprio manual prático de regras para tomada de decisão.
 - Crie uma árvore de decisão usando SPRINT (índice Gini). Você pode criá-la usando cálculos manuais ou uma planilha para realizar os cálculos básicos.
 - Crie outra árvore de decisão, agora usando a abordagem de entropia e ganho de informação (ID3). Você pode usar uma planilha como calculadora nesse exercício.
 - Embora o conjunto de dados apresentado seja extremamente pequeno, tente criar uma pequena rede neural para ele.
 - Use um programa de *software* para árvore de decisão automática (p.ex., See5; faça o *download* de uma versão de teste em rulequest.com) para criar uma árvore para os mesmos dados.
 - Registre as predições das últimas três observações de cada uma das cinco abordagens de classificação.
 - Comente semelhanças e diferenças das abordagens. O que você aprendeu nesse exercício?
5. Você trabalhou com redes neurais e outras técnicas de *data mining*. Dê exemplos de onde cada uma delas foi usada. Com base no seu conhecimento, como você diferenciaria essas técnicas? Suponha que daqui a alguns anos você se depare com uma situação na qual rede neural ou outras técnicas de *data mining* poderia ser usada para criar uma aplicação interessante para sua empresa. Você possui um *trainee* trabalhando com você para fazer o trabalho pesado. Como você decidiria se a aplicação é apropriada para uma rede neural ou para outro modelo de *data mining*? Baseado nas suas tarefas escolares, qual orientação específica de *software* você daria ao seu *trainee* para que ele/ela seja rapidamente produtivo para você? Sua resposta

para essa questão deve mencionar o *software* específico, descrever como empreender a definição do modelo/rede neural e validar a aplicação.

Exercícios na internet

- Investigue os *websites* de vários fornecedores de rede neural, como California Scientific Software (calsci.com), NeuralWare (neuralware.com) e Ward Systems Group (wardsystems.com) e analise alguns de seus produtos. Faça o *download* de pelo menos dois *demos* e instale, execute e compare-os.
- Existe um bom repositório de dados que foi usado para testar o desempenho de redes neurais e muitos algoritmos de aprendizado de máquina. Esse repositório pode ser acessado em ics.uci.edu/~mllearn/MLRepository.html. Alguns conjuntos de dados são realmente destinados para testar os limites de algoritmos atuais de aprendizado de máquina e comparar seu desempenho em oposição a novas abordagens de aprendizado. Entretanto, alguns conjuntos de dados menores podem ser úteis na exploração da funcionalidade do *software* que você deve baixar para o Exercício na internet 1 ou do *software* disponível como complemento deste livro, como o STATISTICA DataMiner. Faça o *download* de pelo menos um conjunto de dados do repositório da UCI (p.ex., Bancos de dados de verificação de crédito, Banco de dados imobiliário). Em seguida, aplique redes neurais e métodos de árvore de decisão, conforme necessário. Prepare um relatório com seus resultados. (Alguns desses exercícios também poderiam ser feitos em grupo ou até mesmo serem propostos como projetos semestrais para um trabalho final etc.)
- Acesse calsci.com e leia sobre diversas aplicações comerciais. Prepare um relatório que resuma as aplicações.
- Acesse nd.com. Leia sobre as aplicações da empresa em investimentos e negociação. Prepare um relatório sobre as aplicações.
- Acesse nd.com. Faça o *download* da versão de teste do Neurosolutions para Excel e faça uma experiência com ele, usando um dos conjuntos de dados dos exercícios deste capítulo. Prepare um relatório sobre sua experiência com a ferramenta.
- Acesse neoxi.com. Identifique pelo menos duas ferramentas de *software* que não foram mencionadas neste capítulo. Visite os *sites* e prepare um breve relatório sobre os recursos dessas ferramentas.
- Acesse neuroshell.com. Veja os exemplos em Gee Whiz. Comente a viabilidade de atingir os resultados afirmados pelos desenvolvedores desse modelo de rede neural.

8. Acesse **easynn.com**. Faça o *download* da versão de teste do *software*. Após instalá-lo, encontre o arquivo de amostra chamado *Houseprices.tvq*. Treine a rede neural novamente e teste o modelo fornecendo alguns dados. Prepare um relatório sobre sua experiência com esse *software*.
9. Acesse **statsoft.com**. Vá até Downloads e baixe pelo menos três documentos informativos sobre aplicações. Qual dessas aplicações pode ter usado redes neurais?
10. Acesse **neuralware.com**. Prepare um relatório sobre os produtos oferecidos pela empresa.

CASO DE APLICAÇÃO DE FINAL DO CAPÍTULO

Notas de risco de crédito soberano usando redes neurais

Empresas como a Standard & Poor's Corporation, Moody's Investors Service e Fitch Ratings fornecem indicadores alfabéticos para risco de crédito. No decorrer de um longo tempo, essas notas estavam em uso para avaliar empresas e instituições financeiras. Entretanto, emitir classificações soberanas é relativamente novo, mas tem visto rápida expansão nos últimos anos. O número de soberanias classificadas cresceu de 17, em 1989 para 63, em 1998. Notas de risco de crédito soberano estão recebendo grande importância global, tanto como uma medida de risco de crédito para um país quanto como uma medida das empresas que funcionam dentro de um país. O Bank for International Settlements (**bis.org**) esteve à frente no uso de notas de risco de crédito na determinação da adequação de capital.

Múltiplos fatores são usados na realização da análise do risco de crédito para classificação de países soberanos. Eles incluem índices financeiros; ambiente econômico, político e regulatório; e tendências da indústria. No contexto de modelos quantitativos, usar dados financeiros, econômicos e de negócios para alcançar uma nota de risco de crédito é um processo desafiador devido às interações complexas e não-lineares entre variáveis diferentes. Porém, esse processo de avaliação de risco carece de uma teoria bem definida, a qual dificulta a aplicação de técnicas matemáticas convencionais ou técnicas baseadas em regras, embora existam numerosas abordagens quantitativas.

A RNA é adequada para modelagem dos determinantes das notas de risco de crédito porque não exige especificação prévia de modelos teóricos. Seu poder próprio de classificar resultados serve para produzir uma escala graduada de classificação. A RNA oferece uma alternativa para as abordagens econométricas nas quais não há hipóteses quanto às propriedades e relações subjacentes dentro dos dados. Ela ganha de todos os outros modelos na derivação de significado a partir de dados complicados ou imprecisos. Uma implementação de RNA bem-sucedida gerará um sistema de relações que foi aprendido da observação

de exemplos anteriores, e ela consegue generalizar e aplicar essas lições a exemplos novos.

Bennell et al.(2006) compararam implementações de RNA à abordagem padrão para análise de risco de crédito de probit. O conjunto de amostra incluiu 1.383 observações anuais (fim de calendário do ano) de notas de risco de crédito soberano para moedas estrangeiras a longo prazo, atribuídas por 11 agências de classificação de risco internacionais para 70 devedores soberanos durante o período entre 1989 e 1999. As variáveis de entrada foram escolhidas por estarem em consonância com os fatores enfatizados tanto na literatura teórica quanto na empírica como determinantes da capacidade e disposição de devedores soberanos pagarem a dívida externa.

Alguns dos indicadores econômicos que foram escolhidos como variáveis explicativas são:

Variável de entrada	Descrição
Dívida externa/exportação	Dívida externa total relativa às exportações para o ano anterior
Balanço fiscal	Déficit (–) ou superávit (+) médio anual do governo federal relativo ao PIB para os três anos anteriores (porcentagem)
Balança externa	Saldo médio anual de conta corrente relativo ao PIB para os três anos anteriores (porcentagem)
Índice de inflação	Taxa média anual de inflação de preços ao consumidor para os três anos anteriores (porcentagem)
PIB per capita	PIB para o ano anterior (em dólares americanos)
Crescimento do PIB	Crescimento real médio anual do PIB em uma base ano sobre ano para os quatro anos anteriores (porcentagem)

Indicador de desenvolvimento	Classificação do país pelo Fundo Monetário Internacional para o ano atual (1 = industrial, 0 = não industrial)
------------------------------	--

Fonte: adaptado de J. Bennell, D. Crabbe, S. Thomas e O. Gwilym, "Modelling Sovereign Credit Ratings: Neural Networks Versus Ordered Probit", *Expert Systems with Applications*, Abril 2006, pp.415-425.

Além das variáveis macroeconômicas especificadas aqui, dois conjuntos de variáveis indicadoras foram incluídos para captar efeitos em classificações soberanas em um determinado ano; classificação soberana atribuída por outras agências de classificação de risco e pela localização da soberania em uma região geográfica específica.

Os dados foram divididos em três grupos: treinamento (na amostra), teste (fora da amostra) e validação cruzada. A divisão visada dos dados em 65%, 20% e 15% foi estabelecida para treinamento, teste e validação cruzada, respectivamente. Entretanto, a separação dos dados foi limitada pelas 16 repetições de notas por diferentes agências. Foram implementadas redes progressivas com multicamadas, todas com uma camada escondida. O número de neurônios na camada escondida foi otimizado pela adição sequencial de neurônios extras até que nenhuma melhoria na classificação fora da amostra fosse obtida.

Os autores usaram taxas de aprendizado e valores de *momentum* diferentes, variando de 0,7 a 1. O treinamento foi realizado por números diferentes de ciclos (chamados *épocas*): 1.000, 2.000, 3.000, 4.000 e 5.000. Ao comparar o erro médio absoluto ao longo dos diferentes testes, os autores selecionaram uma rede progressiva generalizada (GFF) como a rede de melhor desempenho.

Vários critérios são importantes na avaliação do desempenho das redes neurais. É importante distinguir entre precisão de modelagem dentro da amostra e precisão preditiva fora da amostra. Critérios informativos adicionais também estão incluídos: porcentagem corretamente classificada dentro de dois ou três graus de nota, desvio máximo da nota correta e erro médio absoluto. Os modelos de rede neural foram testados várias vezes, e os autores relataram o desempenho médio, assim como o melhor desempenho em cada critério de desempenho.

As agências de classificação avaliam coletivamente as classificações soberanas para moeda estrangeira através da avaliação de fatores consistentes em conjunto com aqueles enfatizados pela teoria como vitais para determinar a capacidade e disposição da soberania para pagar a dívida externa. No caso de modelos de classificação e regressão, o melhor modelo foi obtido do treinamento para 5.000 épocas. Mantendo em men-

te a porcentagem de notas classificadas com precisão, o modelo de rede neural baseado em classificação desempenha seu melhor em 42,4% de acertos, com um desempenho médio de 40,4%, seguido pelo modelo de rede neural baseado em regressão, com 33,9% e 34,6% para melhor desempenho e desempenho médio, respectivamente. Notas classificadas corretamente dentro de um grau foram obtidas em 67,3% e 73,5% dos casos e um desempenho médio em 63,6% e 68,9% dos casos para os modelos de classificação e regressão, respectivamente. Dentro de três graus, o modelo de rede neural baseado em regressão classificou precisamente uma média de 96,7% das notas, com os outros dois modelos chegando à marca de 90% de precisão.

Na comparação dos dois modelos de rede neural, o modelo de regressão obtém uma porcentagem menor de notas corretamente classificadas do que o modelo de classificação, mas tende a se desviar muito menos se uma nota não estiver precisamente correta.

As descobertas indicam que a RNA com o intuito de adaptar notas de risco de crédito para corporações conforme feito pelas maiores agências de classificação de risco (p.ex., Moody's, S&P) pode ser aplicada com êxito às classificações soberanas. O papel de um analista e o processo parcialmente subjetivo de avaliar notas de risco de crédito não podem ser eliminados pelas redes neurais. Porém, parece que a RNA poderia informar e dar suporte ao analista no processo de tomada de decisão.

Fontes: J. Bennell, D. Crabbe, S. Thomas e O. Gwilym, "Modelling Sovereign Credit Ratings: Neural Networks versus Ordered Probit", *Expert Systems with Applications*, April 2006, pp.415-425; e S. Hoti e M. McAleer, *Country Risk Ratings: An International Comparison*, e-u-tokyo.ac.jp/cirje/research/papers/mcaleer/mcaleer4.pdf (acessado em Março de 2006).

QUESTÕES SOBRE O CASO

1. O que são classificações soberanas? Por que elas são importantes?
2. Qual é o papel desempenhado pelas agências de classificação de risco?
3. Qual é a função das redes neurais nas classificações soberanas? Você acha que devemos depender completamente da predição das redes neurais?
4. O que você concluiria a partir dos resultados obtidos nas experiências de predição de nota baseada em rede neural?
5. Você é um analista de crédito na Standard & Poor's. Foi solicitado que você classifique o crédito soberano da Índia. Quais fatores você levaria em consideração para chegar a uma nota de risco de crédito, e como você usaria redes neurais para chegar a um resultado? Explique.

Referências

- Ainscough, T.L. e J.E. Aronson. (1999). "A Neural Networks Approach for the Analysis of Scanner Data." *Journal of Retailing and Consumer Services*, Vol.6.
- Altman, E.I. (1968). "Financial Ratios, Discriminant Analysis and the Prediction of Corporate Bankruptcy." *Journal of Finance*, Vol.23.
- Bennell, J., D. Crabbe, S. Thomas e O. Gwilym. (2006, April). "Modelling Sovereign Credit Ratings: Neural Networks versus Ordered Probit", *Expert Systems with Applications*.
- Collard, J.E. (1990). "Commodity Trading with a Neural Net." *Neural Network News*, Vol.2, No.10.
- Collins, E., S. Ghosh, C.L. e Scofield. (1988). "An Application of a Multiple Neural Network Learning System to Emulation of Mortgage Underwriting Judgments", *IEEE International Conference on Neural Networks*.
- Davis, J.T., A. Episcopos e S. Wettimuny. (2001). "Predicting Direction Shifts on Canadian-U.S. Exchange Rates with Artificial Neural Networks", *International Journal of Intelligent Systems in Accounting, Finance and Management*, Vol.10, No.2.
- Dutta, S. e S. Shakh. (1988, July 24-27). "Bond-Rating: A Non-Conservative Application of Neural Networks", *Proceedings of the IEEE International Conference on Neural Networks*, San Diego.
- Fadlalla, A. e C. Lin. (2001). "An Analysis of the Applications of Neural Networks in Finance." *Interfaces*, Vol.31, No.4.
- Fishman, M., D. Barr e W. Loick. (1991, April). "Using Neural Networks in Market Analysis", *Technical Analysis of Stocks and Commodities*.
- Fozzard, R., G. Bradshaw e L. Ceci. (1989). "A Connectionist Expert System for Solar Flare Forecasting", em D. S. Touretsky (ed.), *Advances in Neural Information Processing Systems* Vol.1. San Mateo, CA: Kaufman Publishing.
- Francett, B. (1989, January). "Neural Nets Arrive." *Computer Decisions*.
- Gallant, S. (1988, February). "Connectionist Expert Systems", *Communications of the ACM*, Vol.31, No.2.
- Haykin, S.S. (1999). *Neural Networks: A Comprehensive Foundation*, 2nd ed. Upper Saddle River, NJ: Prentice Hall.
- Hill, T., T. Marquez, M. O'Connor e M. Remus. (1994). "Neural Network Models for Forecasting and Decision Making", *International Journal of Forecasting*, Vol.10.
- Hopfield, J. (1982, April). "Neural Networks and Physical Systems with Emergent Collective Computational Abilities." *Proceedings of National Academy of Science*, Vol.79, No.8.
- Hopfield, J.J. e D.W. Tank. (1985). "Neural Computation of Decisions in Optimization Problems", *Biological Cybernetics*, Vol.52.
- Kamijo, K. e T. Tanigawa. (1990, June 7-11). "Stock Price Pattern Recognition: A Recurrent Neural Network Approach", *International Joint Conference on Neural Networks*, San Diego.
- Lee, P.Y., S.C. Hui e A.C.M. Fong. (2002, September/October). "Neural Networks for Web Content Filtering." *IEEE Intelligent Systems*.
- Liang, T.P. (1992). "A Composite Approach to Automated Knowledge Acquisition." *Management Science*, Vol.38, No.1.
- McCulloch, W.S. e W.H. Pitts. (1943). "A Logical Calculus of the Ideas Imminent in Nervous Activity." *Bulletin of Mathematical Biophysics*, Vol.5.
- Mighell, D. (1989). "Back-Propagation and Its Application to Handwritten Signature Verification", em D.S. Touretsky (ed.), *Advances in Neural Information Processing Systems*. San Mateo, CA: Kaufman.
- Minsky, M. e S. Papert. (1969). *Perceptrons*. Cambridge, MA: MIT Press.
- Principe, J.C., N.R. Euliano e W.C. Lefebvre. (2000). *Neural and Adaptive Systems: Fundamentals Through Simulations*. New York: Wiley.
- Rochester, J. (ed.). (1990, February). "New Business Uses for Neurocomputing." *I/S Analyzer*.
- Surkan, A. e J. Singleton. (1990). "Neural Networks for Bond Rating Improved by Multiple Hidden Layers." *Proceedings of the IEEE International Conference on Neural Networks*, Vol.2.
- Tang, Z., C. de Alameda e P. Fishwick. (1991). "Time-Series Forecasting Using Neural Networks vs. Box-Jenkins Methodology." *Simulation*, Vol.57, No.5.
- Thaler, S.L. (2002, January/February). "AI for Network Protection: LITMUS: - Live Intrusion Tracking via Multiple Unsupervised STANNOS." *PC AI*.
- Walczak, S., W.E. Pofahi e R.J. Scorpio. (2002). "A Decision Support Tool for Allocating Hospital Bed Resources and Determining Required Acuity of Care." *Decision Support Systems*, Vol.34, No.4.
- Wilson, R. e R. Sharda. (1994). "Bankruptcy Prediction Using Neural Networks." *Decision Support Systems*, Vol.11.
- Zahedi, F. (1993). *Intelligent Systems for Business: Expert Systems with Neural Networks*. Belmont, CA: Wadsworth.