

Relatório Técnico

**Núcleo de
Computação Eletrônica**

- NEUROLAB - Um Ambiente Cliente-Servidor para Desenvolvimento e Aplicação de Redes Neurais Artificiais

**Antonio Carlos Gay Thomé
Marcello Baptista de Martino
Liliane Videira Assaf**

NCE - 16/99

Universidade Federal do Rio de Janeiro

NEUROLAB - Um Ambiente Cliente-Servidor para Desenvolvimento e Aplicação de Redes Neurais Artificiais

Antonio Carlos Gay Thomé

Universidade Federal do Rio de Janeiro - UFRJ
IM / Departamento de Computação
Núcleo de Computação Eletrônica
thome@nce.ufjf.br - 55 21 598-3163

Marcello Baptista de Martino

Liliane Videira Assaf
Centro de Pesquisa de Energia Elétrica - CEPTEL
martino@fund.cepel.br
55-21-598-2138

Resumo

Trata-se de uma ferramenta para ambiente Windows, cuja finalidade é a de facilitar o estudo, o desenvolvimento e o emprego de redes neurais na solução de problemas reais. O NeuroLab, através de uma filosofia cliente-servidor, oferece recursos e facilidades para o emprego integrado de redes neurais e aplicações convencionais na solução de problemas complexos, de uma forma simples, funcional e elegante. A ferramenta, pela sua facilidade de uso, funcionalidade e flexibilidade, tem grandes possibilidades de agradar tanto os especialistas no assunto, como as comunidades acadêmica e empresarial e, inclusive, os iniciantes.

Introdução

Apesar das dificuldades e das restrições tecnológicas ainda grandes sobre a trajetória das redes neurais, constata-se hoje uma presença já considerável e um interesse ainda crescente, tanto da comunidade acadêmica como da sociedade em geral, no seu estudo e no emprego de suas técnicas na resolução de problemas reais.

Na área acadêmica, a nível de pós-graduação, há anos que o estudo e a pesquisa sobre diferentes técnicas e modelos neurais recebe a atenção de um grande contingente de alunos e professores. A nível de graduação é notório o aumento do número de cursos que oferecem uma ou mais disciplinas abordando o tema de forma específica, que também é bastante usado para trabalhos de iniciação científica.

Cursos de pós-graduação "lato sensu", mesmo em diferentes áreas do saber, também vêm apresentando um crescente interesse pelo assunto e, principalmente, pela potencialidade de seu emprego na solução de problemas complexos e ainda carentes de solução automatizada.

Constata-se também que a medida em que as técnicas neurais deixam de ser ilustres desconhecidas, cresce a demanda comercial por produtos que a tenham como base ou como parte integrante da solução.

Como reflexo de um tal contexto, observa-se pesquisadores, alunos, e mesmo profissionais de mercado, oriundos das mais distintas

áreas do conhecimento, interessados no conhecimento das redes neurais. Como consequência abre-se a oportunidade para o aparecimento de ferramentas que busquem facilitar o aprendizado e o uso de tais técnicas.

O NeuroLab foi concebido exatamente nesta direção, ou seja, de ser um ambiente para facilitar o projeto e o desenvolvimento de redes neurais e tornar o mais simples e transparente possível o seu emprego e sua integração com qualquer programa ou aplicação convencional.

Nas próximas seções deste artigo serão destacados: a filosofia do NeuroLab e a funcionalidade que este oferece aos seus usuários; seus principais componentes - o SiReNe - que oferece um ambiente amigável e integrado para criação, preparação e validação de redes neurais de diferentes modelos e arquiteturas; e - o SeReNa - que se encarrega de disponibilizar as redes geradas pelo SiReNe para serem utilizadas, na forma cliente-servidor, por uma aplicação convencional qualquer. Será também feita uma simulação do emprego do NeuroLab e finalmente serão citados alguns tópicos para continuidade dos trabalhos.

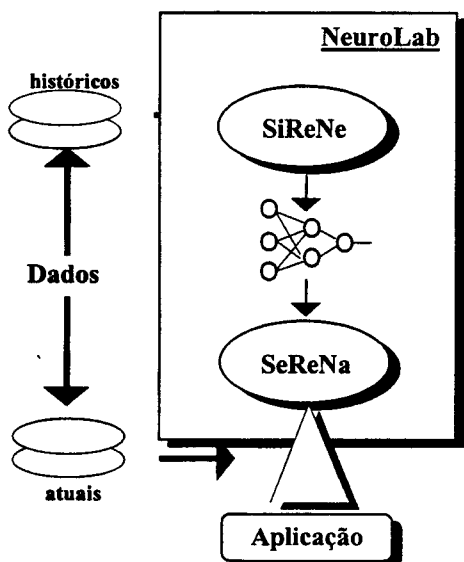
NeuroLab

O NeuroLab é constituído por duas ferramentas independentes, SiReNe e SeReNa, que, utilizadas em fases distintas, permitem a construção de uma rede neural para solução de um determinado problema específico e a colocação desta rede em disponibilidade, de forma

transparente, para utilização pelo sistema aplicativo do usuário final.

O SiReNe é a ferramenta que viabiliza a construção, o treinamento e a validação da rede neural na resolução do problema específico.

O SeReNa é a ferramenta que disponibiliza a interface para a aplicação cliente interagir com a rede neural, criada pelo SiReNe, de modo transparente e simples.



É dispensável citar aqui toda a gama de aplicativos e utilitários desenvolvidos para a plataforma Windows, assim, o NeuroLab foi desenvolvido para esta plataforma com vistas a conseguir se integrar ao maior número de aplicações possível.

A ferramenta SiReNe sendo feita para Windows, se torna um simulador muito fácil de ser usado, pois sua interface segue a interface padrão dos sistemas para Windows, formada por janelas, menus, controles como botões, barras de rolagem, tudo muito conhecido pelos usuários desse ambiente.

A ferramenta SeReNa, sendo também um sistema para Windows, se torna muito útil para os desenvolvedores das aplicações, já que a maioria destas aplicações são hoje desenvolvidas para ambiente Windows.

A linguagem escolhida para a implementação do NeuroLab foi o Visual Basic, por oferecer paradigmas modernos de programação, ser de fácil aprendizado e permitir rápido desenvolvimento de aplicações para Windows. O VB possui *late binding* e oferece facilidades

para construção de aplicações MDI (Multiple Documents Interface) e servidores OLE.

Late Binding é a alocação de memória em tempo de execução da aplicação. Ou seja, a alocação de memória para uma variável não é realizada previamente, e sim, na primeira referência àquela variável durante a execução da aplicação.

Este tipo de alocação de memória é muito importante para a implementação do simulador, já que as matrizes de pesos das redes neurais podem ser de diversos tamanhos. Se a linguagem não oferecesse tal facilidade, teríamos que alocar previamente um espaço de memória suficiente para armazenar estas matrizes. Isso poderia acarretar um desperdício de memória, no caso das matrizes virem a ter tamanhos menores que o reservado para elas, ou poderia impossibilitar a execução do simulador, caso o espaço reservado fosse insuficiente para a alocação destas matrizes. Em linguagens que não possuam esta característica, outros mecanismos, como alocação dinâmica de memória podem ser utilizados para o mesmo fim.

Existe um tipo de aplicação para Windows capaz de manipular documentos. Este tipo de aplicação é conhecido como MDI. Uma aplicação MDI é aquela que permite abrir vários documentos (do mesmo tipo) ao mesmo tempo e que controla todos os documentos que estão abertos. A ferramenta SiReNe foi implementada como sendo uma aplicação MDI.

Existe um tipo de aplicação para Windows com facilidades para comunicação com outras aplicações Windows. Este tipo de aplicação é conhecida como Servidor OLE. Os Servidores OLE são aplicações que provêem objetos, com seus métodos e propriedades, para outras aplicações Windows. A ferramenta SeReNa foi implementada como um Servidor OLE.

A ferramenta SiReNe, por ser MDI, permite ao usuário manipular cada simulação de rede como um documento distinto e criar um arquivo com a descrição da rede neural capaz de resolver o problema. E a ferramenta SeReNa, por ser um Servidor OLE, permite ao usuário utilizar este arquivo em sua aplicação sem conhecer detalhes sobre seu formato.

SiReNe

Trata-se de um Simulador de Redes Neurais para ambiente Windows, que oferece recursos e facilidades para criar, configurar,

treinar, testar e salvar redes neurais, para posterior utilização por aplicações convencionais desenvolvidas para resolverem problemas específicos nas mais diversas áreas do saber.

A facilidade de interação, aliada a flexibilidade oferecida pelos recursos disponíveis, fazem do SiReNe um simulador versátil e com grandes possibilidades de aceitação tanto por especialistas na área como por iniciantes.

Através de menus e de janelas gráficas, o SiReNe oferece ao usuário recursos e facilidades para preparar uma rede neural para a solução de um problema específico qualquer.

Três modelos de rede: Elman, Jordan e MLP (backpropagation) estão atualmente disponíveis para escolha na barra de ferramentas. Após a escolha do modelo, três ambientes são oferecidos: um para definir a arquitetura da rede (número de camadas, número de elementos de processamento por camada e tipo de função de ativação), outro para selecionar os dados a serem utilizados no treinamento e no teste da rede e, finalmente outro, para definir os parâmetros e executar o treinamento ou o teste. O usuário tem ainda a opção de salvar as condições correntes para continuar o treinamento em uma outra ocasião ou salvar a rede para posterior utilização de dentro de uma aplicação convencional qualquer.

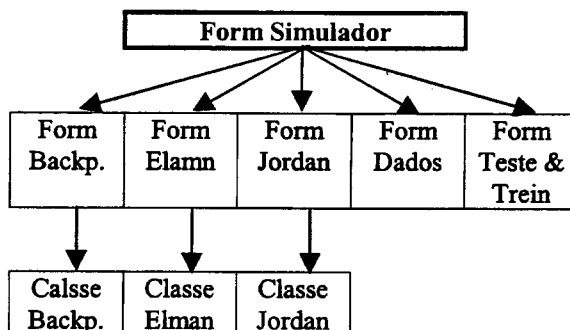
No ambiente de treinamento existem facilidades e recursos para manipular, mesmo de forma interativa durante a execução do treinamento, alguns parâmetros tais como: taxa de aprendizado; momento; percentual de ruído adicional; frequência de apresentação gráfica da evolução do erro; número de ciclos (passos) de treinamento; erro máximo para parada; etc. A qualquer instante o treinamento pode ser suspenso ou interrompido e, neste caso, reinicializado a partir de um outro ponto aleatoriamente selecionado ou do ponto onde estava.

Como todos os sistemas desenvolvidos para Windows, o SiReNe é capaz de manipular um determinado tipo de documento. Por exemplo, o Word manipula textos, o Excel planilhas e o SiReNe redes neurais.

O SiReNe é uma aplicação MDI (Multiple Document Interface). Em sua janela principal (MDI Form) ele oferece um menu e uma barra de ferramenta com opções de criar, abrir, salvar e organizar simulações. Os modelos atualmente disponíveis estão implementados em três janelas filhas (MDI Child Form): backpropagation(MLP), Elman e Jordan. Duas janelas

do tipo "Form Main" se encarregam de oferecer as facilidades para manipulação dos dados e da execução do treinamento e do teste.

Para cada modelo de rede existe uma classe onde estão definidos os procedimentos e funções necessários para a utilização do respectivo modelo neural.

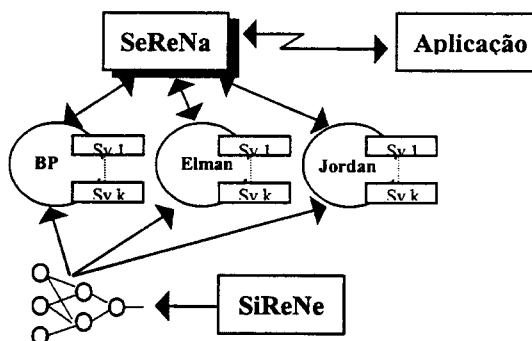


A implementação modular do SiReNe possibilita a adição de novos modelos neurais de forma bastante simples, bastando para isto criar uma MDI Child Form e uma Classe para cada novo modelo.

SeReNa

É a parte do Neurolab que se encarrega de disponibilizar a rede neural gerada pelo SiReNe para ser usada por uma aplicação convencional qualquer, sob a forma cliente-servidor.

O SeReNa oferece métodos (serviços) de manipulação de uma rede neural. Para fazer uso destes métodos, o desenvolvedor da aplicação convencional precisa saber o modelo de rede que deseja empregar e a sintaxe dos respectivos comandos para se comunicar com o servidor.



Os seguintes serviços estão disponíveis no SeReNa: load() - para carregar a rede gerada pelo SiReNe; Processa() - para executar a rede no processamento de um vetor de entrada (observação); DataInput() - para fornecer um vetor

de entrada à rede; DataOutput() - para pegar o resultado gerado pela rede.

Quando uma aplicação que faz uso do SeReNa estiver em execução e fizer a chamada de algum serviço do mesmo, o servidor é automaticamente ativado e sua janela de apresentação é colocada na tela do monitor.

Uso do NeuroLab - Um Exemplo

O objetivo é mostrar como o SiReNe e o SeReNa juntos podem auxiliar o desenvolvimento de uma aplicação convencional qualquer, que precise fazer uso do paradigma neural como parte da estratégia de resolução de um determinado problema.

A fim de melhor mostrar a capacidade e o desempenho do SiReNe na preparação do modelo neural e da facilidade de integração do SeReNa com a aplicação do usuário, optou-se por usar um problema real, razoavelmente complexo e com um conjunto de dados relativamente extenso.

Na busca de um caso real que atendes-se a tais pré-requisitos, decidiu-se avaliar alguns dos problemas oferecidos pelo Proben1, que é uma fonte de problemas reais, próprios para serem resolvidos com o uso das técnicas Neurais. Esta fonte está disponível através da Internet no endereço:

<http://www.cse.cuhk.edu.hk/pub/proben1>

Dentre os diversos problemas existentes, o diagnóstico do câncer foi o escolhido. Ele consiste em diagnosticar o tumor como sendo maligno ou benigno com base em um conjunto de informações laboratoriais sobre as células do organismo doente.

O conjunto de dados a respeito do problema está disponível a qualquer pessoa e pode ser obtido através do endereço:

<http://www.cse.cuhk.edu.hk/pub/proben1/cancer>

Neste endereço existem vários arquivos disponíveis, dentre eles o *cancer.raw*, que foi o utilizado para esta simulação. Neste conjunto de dados encontram-se 699 exemplos, cada um com 10 entradas e 1 saída. Cada entrada, exceto a primeira, que contém apenas um código numérico, representa o resultado de análises laboratoriais feitas sobre as células de um tumor. A saída é a classificação do tumor, como sendo maligno ou benigno, resultante do diagnóstico

real. As entradas possuem valores entre 1 e 10 e as saídas só podem ser 2 (benigno) ou 4 (maligno).

Entradas / Saídas	Significado	Valores
e1	Código numérico	Inteiro
e2	Espessura da massa de células	1 a 10
e3	Uniformidade do tamanho da célula	1 a 10
e4	Uniformidade do formato da célula	1 a 10
e5	Adesão marginal	1 a 10
e6	Tamanho de uma célula epitelial	1 a 10
e7	Núcleo vazio	1 a 10
e8	Cromatina branda	1 a 10
e9	Nucléolo normal	1 a 10
e10	Mitose	1 a 10
s1	Maligno ou Benigno	2 ou 4

A partir da obtenção dos dados, o NeuroLab passa a ser usado na construção da rede neural. A construção e o treinamento da rede são feitos com o auxílio do SiReNe.

Uma fase de preparação e normalização dos dados é importante para obtenção de um melhor desempenho e melhores resultados finais. Foi assim, realizado um tratamento nos dados originais do problema, e um novo conjunto de dados foi gerado. Neste tratamento foram realizadas as seguintes operações:

1. Eliminação dos exemplos (linhas) que tinham alguma entrada ou saída com valor indeterminado, como o exemplo 24.
2. Eliminação da primeira entrada (coluna), que é apenas um código numérico.
3. Divisão por 10 de todos os valores das entradas restantes.
4. Subtração por 3 de todos os valores das saídas.

Com esse tratamento, o novo conjunto de dados passou a ter 683 exemplos, as entradas ficaram limitadas a valores entre 0.1 e 1, e as saídas a valores -1 (benigno) e +1 (maligno).

Para treinamento da rede neural não foram utilizados todos os dados do problema, mas apenas os 99 primeiros exemplos. E, para teste, foram utilizados todos os dados, ou seja, todo o conjunto de 683 exemplos.

O SiReNe foi usado para criar e preparar a rede neural adotando-se heurísticamente, a seguinte arquitetura:

- rede MLP com três camadas
- 9 neurônios na camada de entrada
- 18 neurônios na camada intermediária
- 1 neurônio na camada de saída
- função de ativação tipo *Sigmóide Bipolar* entre as camadas de entrada e intermediária
- função de ativação *Sigmóide Bipolar* entre as camadas intermediária e de saída

Para o treinamento da rede foram escolhidos os seguintes parâmetros:

- tamanho da epoch = 99
- momento = 0.6
- coeficiente de aprendizado = 0.9
- ruído = 0
- erro mínimo = 0.0001

No treinamento a rede alcançou o erro de 0.0202058 com 249 passos. Apesar do erro não ser menor ou igual ao erro mínimo estabelecido como parâmetro de treinamento (0.0001), os resultados obtidos no teste foram satisfatórios e, assim, decidiu-se encerrar a fase de treinamento.

A configuração e os pesos da rede treinada foram salvos em um arquivo (treina99.net), para que a mesma pudesse ser posteriormente utilizada via SeReNa. Como resultado do treinamento apenas 2 dos 55 exemplos benignos foram classificados como malignos, e todos os 44 exemplos malignos foram classificados corretamente.

No entanto para avaliar a capacidade de generalização da rede, utilizamos os 584 exemplos restantes.

Para valores de saída menores que -0.5, considerou-se que a rede forneceu diagnóstico Benigno. Para valores de saída maiores que +0.5, considerou-se diagnóstico Maligno, e os erros que tiveram valores no intervalo de -0.5 a +0.5 exclusive, considerou-se que a rede ficou em dúvida. Na tabela abaixo, que representa a matriz de confusão, vê-se o resultado alcançado com os dados de teste.

	Benignos	Malignos
Benignos	366	1
Não Identificados	2	-
Malignos	21	194

Dos 584 casos testados, apenas 2 não foram identificados pela rede, somente 1 caso

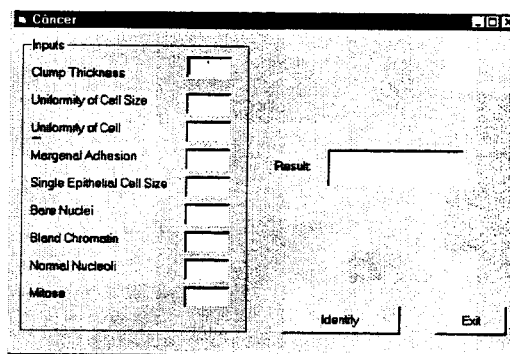
benigno foi erroneamente classificado como maligno e 21 malignos foram classificados como benigno, o que representa uma taxa de acerto de 95.9%.

Isso mostra que a rede conseguiu resolver o problema de diagnóstico do Câncer, obtendo um excelente resultado na generalização dos casos que ela não treinou.

Com a rede configurada e pronta para tratar o problema, resta ver a forma de integrá-la a aplicação final via a ferramenta SeReNa.

Projetou-se uma aplicação bastante simples, já que o objetivo maior é mostrar a facilidade de integração com o SeReNa. O programa desenvolvido para a aplicação foi construído em Visual Basic, é importante destacar no entanto, que ele poderia ter sido desenvolvido em qualquer outra linguagem capaz de manipular recursos Windows.

A aplicação foi criada para fornecer o diagnóstico sobre um tumor cancerígeno, com base nas informações laboratoriais fornecidas pelo usuário. A figura abaixo mostra a interface construída para o diálogo com o usuário. Através dela ele fornece os dados e recebe o diagnóstico.



No quadro “Entradas” estão os campos onde o usuário deve entrar com todas a informações necessárias para a análise do tumor. O botão “Identifica” dispara o processo de análise do tumor fornecendo, ao final desse processo, o seu diagnóstico que pode ser “Benigno”, “Maligno” ou “Não Identificado”, caso a rede não tenha conseguido identificar o tipo de tumor.

A aplicação, para proceder o diagnóstico do tumor, faz uso da rede neural previamente preparada pelo SiReNe e disponibilizada pelo SeReNa. A sintaxe a ser observada pelo construtor da aplicação é a seguinte:

1 - Na parte de declaração das variáveis da janela da aplicação, ele deve declarar uma variável objeto do tipo Backpropagation:

Private RedeCancer As New SeReNa.classBP3

2 - No evento de inicialização da janela da aplicação, ele precisa carregar a rede já treinada no objeto previamente criado, utilizando para isto o método *Load*:

```
arqRede = "treina99.net"  
RedeCancer.Load (arqRede)
```

3 - No evento "*click*" do botão "Identify", os dados de entrada devem ser tratados, e o valor de cada uma das entradas da rede deve ser especificado, utilizando para isto o método *DataInput*.

```
RedeCancer.DataInput(1) = CInt(e1.Text) / 10  
RedeCancer.DataInput(2) = CInt(e2.Text) / 10  
RedeCancer.DataInput(3) = CInt(e3.Text) / 10  
RedeCancer.DataInput(4) = CInt(e4.Text) / 10  
RedeCancer.DataInput(5) = CInt(e5.Text) / 10  
RedeCancer.DataInput(6) = CInt(e6.Text) / 10  
RedeCancer.DataInput(7) = CInt(e7.Text) / 10  
RedeCancer.DataInput(8) = CInt(e8.Text) / 10  
RedeCancer.DataInput(9) = CInt(e9.Text) / 10
```

A seguir deve ser passado para o SeReNa, o comando para processar as entradas através da rede, o que é feito por intermédio do método *Run*.

RedeCancer.Run

Após o processamento da rede, o resultado gerado deve ser buscado através do método *DataOutput*.

```
s1 = RedeCancer.DataOutput(1)
```

Finalmente, o programa aplicativo deve tratar a resposta recebida da rede e, no caso, fornecer o diagnóstico do tumor ao usuário.

```
If Abs(1 - s1) <= 0.5 Then  
    saida.ForeColor = vbRed  
    saida.Text = "MALIGNO"  
ElseIf Abs(1 - s1) < 0.5 Then  
    saida.ForeColor = vbGreen  
    saida.Text = "BENIGNO"  
Else  
    saida.ForeColor = vbYellow  
    saida.Text = "NÃO IDENTIFICADO"  
End If
```

Embora o código aqui apresentado seja de uma aplicação em VisualBasic, esta mesma aplicação poderia estar escrita do mesmo modo em VisualC++, Delphi ou PowerBuilder. Assim, com uma sintaxe bastante simples e de fácil implementação, o implementador de qualquer aplicação convencional passa a poder fazer uso de técnicas neurais como estratégia na solução de um determinado problema específico.

Conclusões

O NeuroLab é um ambiente de suporte ao desenvolvimento de aplicações para ambiente Windows, que apresentem a necessidade de usar redes neurais como parte da estratégia de resolução de um determinado problema.

O ambiente NeuroLab oferece duas ferramentas, uma para criação de redes neurais, o SiReNe, e outra, o SeReNa, para permitir que estas redes sejam acessadas pelas aplicações finais de forma simples e transparente ao usuário.

Estas características, aliadas à facilidade de utilização de ambas as ferramentas, fazem do NeuroLab um poderoso instrumento de divulgação da técnica de Redes Neurais e de integração desta com as aplicações práticas.

Os próximos passos previstos para este trabalho incluem: introduzir um ambiente para tratamento e análise dos dados; acrescentar novos modelos de rede aos já existentes e, também, ampliar a forma de armazenamento para aceitar também o padrão ODBC.

Bibliografia

Neural Networks: A Comprehensive Foundation, Simon Haykin, IEEE Press

Neural Computing: Theory and Practice, Philip D. Wasserman, Van Nostrand Reinhold

Fundamentals of Artificial Neural Networks, Mohamed H. Hassoun, MIT Press, 1995

Redes Neurais aplicadas à Aquisição de Conhecimento em Sistemas Elétricos de Potência, Marcello Baptista de Martino, COPPE - UFRJ, 1994

Simulador de Redes Neurais, Adriana Moura Viera, DEL - UFRJ, 1998

Microsoft Visual Basic, Microsoft Corp., 1995

Usando Visual Basic 5, M. McKelvy, Campus,
1997