

Ferramenta computacional para implementação de redes neuronais
optimização das capacidades gráficas

Alexandra Oliveira

ao@fe.up.pt

Professor Joaquim Marques de Sá

December 2007

INEB - Instituto de Engenharia Biomédica
FEUP/DEEC, Rua Dr. Roberto Frias, 4200-645 PORTO

Conteúdo

1	Sobre a Ferramenta Computacional	2
2	Acknowledgments	3
3	Principais fragilidades da primeira versão	5
4	Interface gráfica - alterações principais	7
	Bibliography	12

Capítulo 1

Sobre a Ferramenta Computacional

As redes neuronais são uma ferramenta poderosa com aplicações em diversas áreas de conhecimento como por exemplo Medicina, Mercados Financeiros e Indústria.

Uma rede neuronal artificial é um modelo matemático inspirado em redes neuronais biológicas. São constituídas por conjuntos de neurónios com capacidades de processamentos. Estes neurónios estão ligados entre si de forma a que a informação circule entre todos os neurónios. A maior parte das redes neuronais são sistemas adaptativos que alteram a sua estrutura com base em informações internas e externas que circulam pela rede neuronal durante a fase de treino([1]).

As redes neuronais são uma das áreas de estudo do INEB - Instituto de Engenharia Biomédica, em particular do grupo NNIG - Neural Networks Interest Group que pertence à Divisão de Sinal e Imagem. Este grupo tem diversos trabalhos publicados na área e estes trabalhos motivaram o desenvolvimento de um software novo.

Este software, da qual já existe uma primeira versão, tem como objectivo a implementação de algoritmos de redes neuronais tendo uma interface simples e modular em que cada componente representa uma característica distinta da rede neuronal. O grupo já dispõe de uma primeira versão do software desenvolvida em Microsoft Visual C# e que se encontra descrita em [2]. Contudo, esta primeira versão necessitava de ser testada e de uma interface gráfica mais intuitiva de forma a facilitar o acesso aos algoritmos que implementa. O trabalho decorrente desta necessidade é o tema deste relatório.

Capítulo 2

Acknowledgments

O desenvolvimento deste software é suportado pela FCT (Fundação para a Ciência e Tecnologia) e FEDER sob o projecto *POSCEIA569182004*.

Este software foi inspirado no projecto *Neural Network Library Version 0.1 (April 2002)* de *Franck Fleurey*. Este projecto foi distribuído sob a licença de uso: GNU general public license. Este projecto é constituído por duas livrarias: *NeuralNetworkLibrary* e *NeuralNetworkGUI*. Se no caso da livraria *Neural Network Library* apenas foram acrescentadas classes e foram feitas pequenas alterações, o mesmo não aconteceu com a interface que o autor desenvolveu. Esta interface serviu de inspiração de alguns componentes gráficos que estão a ser desenvolvidos.

A livraria *ZedGraph* desenvolvida por John Champion é a livraria usada no nosso software para desenhar gráficos 2D. Esta livraria é formada por um conjunto de classes, escritas em C#, disponibilizando várias opções de gráficos e várias funcionalidades que se revestem de grande importância para o desenvolvimento deste projecto.

Uma caixa de texto que faz a validação de diferentes tipos de representações numérica é também usado neste software e o seu desenvolvimento deve-se a Oscar Bowyer.

Uma outra contribuição de relevo para o nosso software é da autoria de Yossi Rozenberg e consiste no repositórios de diferentes algoritmos que calculam diferentes estatísticas para um dado conjunto de valores.

Dentro da área da estatística falta ainda realçar o trabalho do CenterSpace Software cuja sua contribuição se encontra num conjunto de rotinas que implementam um histograma.

O desenvolvimento deste software contou ainda com a preciosa contribuição de Daniel Carrilho, Paulo Ricca e Adriano Correia ao nível da programação de algumas funcionalidades assim como com a contribuição de todos os membros do NNIG - Neural Networks Interest Group que testaram e deram

sugestões.

Capítulo 3

Principais fragilidades da primeira versão

No decorrer dos diversos testes da primeira versão do software foram detectadas alguns bugs e fragilidades do sistema que necessitavam uma correcção urgente. As principais e prementes eram:

1. a complexidade da organização e de dependência dos projectos dificultando o acesso às suas funcionalidade;
2. ambiente gráfico com componentes grandes dificultando o manuseamento de todas no mesmo ambiente de trabalho;
3. a dependência entre componentes gráficos.

Para simplificar a complexa distribuição das classes contidas no projecto descrito em [2] as classes obsoletas foram eliminadas e as outras reorganizadas em apenas dois projectos ficando a solução como ilustra a figura 3.1. É de notar que as descrições técnicas das classes que compõem esta solução se encontram em [2].

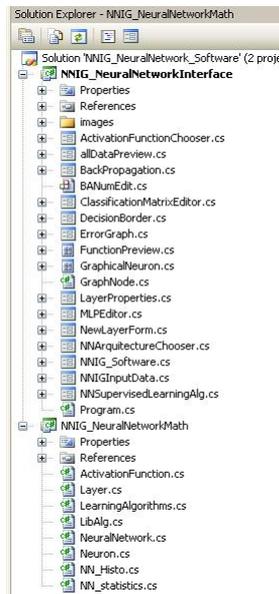


Figura 3.1: Re-distribuição das classes pelo projecto.

Dado que a interface gráfica, que se encontra no projecto **NNIG_NeuralNetworkInterface**, depende dos algoritmos matemático que implementam todas as características de uma rede neuronal e que se podem encontrar no projecto **NNIG_NeuralNetworkMath**, existe nesta solução uma dependência por parte do primeiro em relação ao segundo projecto. Esta dependência encontra-se representada no diagrama 3.2 onde o símbolo \rightarrow significa “depende de”.



Figura 3.2: Dependências entre os projectos da solução NNIG_NN_SOFTWARE

Em relação à versão anterior o projecto **ChartLibrary** e **NeuralNetwork GUI** foram eliminado uma vez que deixaram de satisfazer as necessidades da nossa solução; o projecto **Neural Networks** e **RedesNeuronais_NNIG** foram fundidos de forma a que todas as rotinas de apoio à interface gráfica estivessem no mesmo projecto.

As descrições técnicas de cada classe contida no projecto **NNIG_NeuralNetworkMath** podem ser encontradas em [2].

Capítulo 4

Interface gráfica - alterações principais

Uma das primeiras questões a serem resolvidas foi o facto de que cada modulo da interface gráfica ser de grandes dimensões dificultando a sua disposição no ambiente de trabalho. Assim, todas as componentes foram transformadas em janelas com capacidade de minimização e pertencentes à janela principal do software. Estas janelas são independentes umas das outras podendo no entanto partilhar informações. Por exemplo, o utilizador ao introduzir informações sobre os dados (clicando no botão OK do controlo **InputData**) estas são enviadas para o controlo **MLPEditor**. Esta independência permite que todas as janelas necessárias para uma experiência com redes neuronais artificiais estejam abertas antes de serem introduzidos dados no sistema.

Após terminadas as alterações em cada componente da interface gráfica, o utilizador deve clicar no botão OK para que as suas opções fiquem registadas no núcleo matemático da ferramenta. Esta pequena alteração originou num novo design de algumas componentes, nomeadamente no **MLPEditor** como se pode ver na figura 4.1.

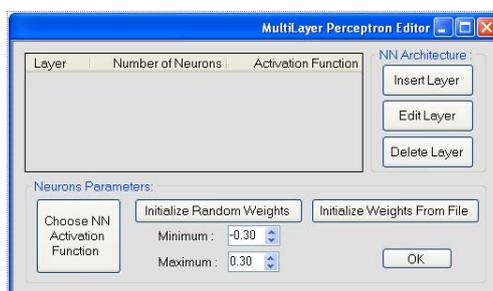


Figura 4.1: Re-organização do control MLP Editor

Como os módulos gráficos desta ferramenta passaram a ser janelas, com todas as funcionalidades de uma janela do Windows, houve a necessidade de introdução de um sistema de segurança contra o

seu fecho accidental. Assim, após cada click num botão de fecho de janela é perguntado ao utilizador se pretende efectivamente abandonar a componente gráfica e se pretende guardar o seu trabalho.

Um outra funcionalidade nova do software em questão é a possibilidade de continuar o treino de uma rede neuronal. Para que esta funcionalidade ficasse disponível aos utilizadores foi acrescentado na barra de ferramentas da janela principal dois botões: o Run e o Continue como ilustra a figura 4.2.



Figura 4.2: Janela principal da ferramenta computacional.

O botão **Run** utiliza para treino todas as definições iniciais introduzidas pelo utilizador e o botão **Continue** inicia o treino a partir das definições obtidas na última iteração da experiência anterior.

A barra de ferramentas da janela principal do software foi também re-estruturada por forma a tornar este software mais geral, nomeadamente no que diz respeito às arquitecturas de rede e aos algoritmos de treino existentes. Assim, foram criados dois novos controlos: o **NNArchitecture** e o **Supervised Learning**. O primeiro controlo é criado quando se clica no botão **NN ARCHITECTURE** e o segundo no botão **SUPERVISED LEARNING** da barra de ferramentas da janela principal. Estes controlos encontram-se representados nas figuras 4.3 e 4.4 respectivamente.

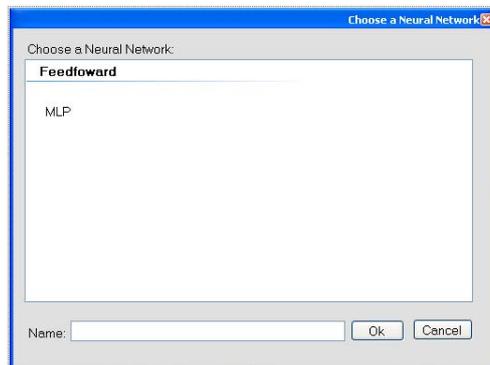


Figura 4.3: Controlo que permite escolher uma arquitectura da rede neuronal.

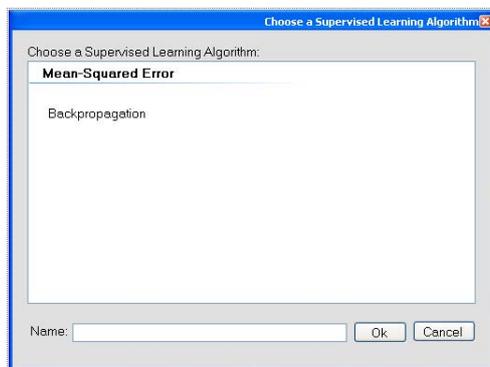


Figura 4.4: Controlo que permite escolher um algoritmo de treino supervisionado.

O controlo representado na figura 4.3, até ao momento, permite apenas criar um controlo do tipo **MLPEditor**. Os detalhes técnicos deste controlo encontram-se descritos no relatório técnico [2].

O controlo representado na figura 4.4, até ao momento, permite apenas criar um controlo do tipo **Backpropagation** cujos detalhes técnicos se encontram descritos no relatório técnico [2].

Uma nova e pequena alteração encontra-se no controlo **Layer Properties**. Agora, neste controlo, e por baixo do botão **Choose Layer Activation Function** é possível visualizar o nome a função de activação escolhida para todos os neurónios daquela camada (4.5).

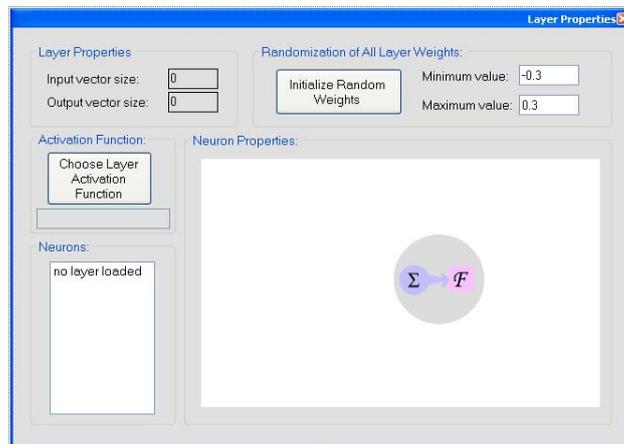


Figura 4.5: Janela que permite visualizar e alterar as propriedades de uma determinada camada de neurónios.

O controlo **NNOutput** foi temporariamente retirado do projecto por forma a se poder estudar uma melhor opção para a visualização das resposta da rede neuronal.

Durante a re-estruturação da interface gráfica desta ferramenta computacional foi também estudada a possibilidade de ligar estas janelas criando uma rede de componentes gráficas. Cada janela deveria ser capaz de identificar as componentes a ela ligada e ser capaz de identificar o sentido de fluxo da informação através desta rede de componentes gráficos. Por isso, foi explorado a possibilidade de implementação de grafos orientados aplicados às componentes gráficas. No entanto, e devido ao excesso de tempo que esta tarefa acarretava, foi abandonada a ideia.

Quanto à estrutura matemática desta nova versão foi alterada o conceito associado à arquitectura de rede. Antes, o número de neurónios na última camada da rede neuronal era igual ao número de classes representadas nos dados, agora e no caso particular de duas classes, apenas consideramos um neurónio.

Uma outra medida de segurança introduzida nesta nova versão diz respeito à validação da dimensão da matriz inicial de pesos quando esta matriz é introduzida no sistema pelo utilizador através de um ficheiro pré-concebido. Após a leitura deste ficheiro o sistema é capaz de verificar se a dimensão destas matrizes são concordantes com a arquitectura da rede definida e no caso de existirem incoerência avisa o utilizador para rever as suas opções.

Para que a visualização da evolução do erro em cada treino ficasse mais clara foi introduzida, no controlo do gráfico do erro, a possibilidade de apagar as curvas lá representadas. Assim, a cada novo treino o utilizador pode optar por apagar a curva já desenhada e ficar apenas com a evolução do

presente treino da rede neuronal (4.6).

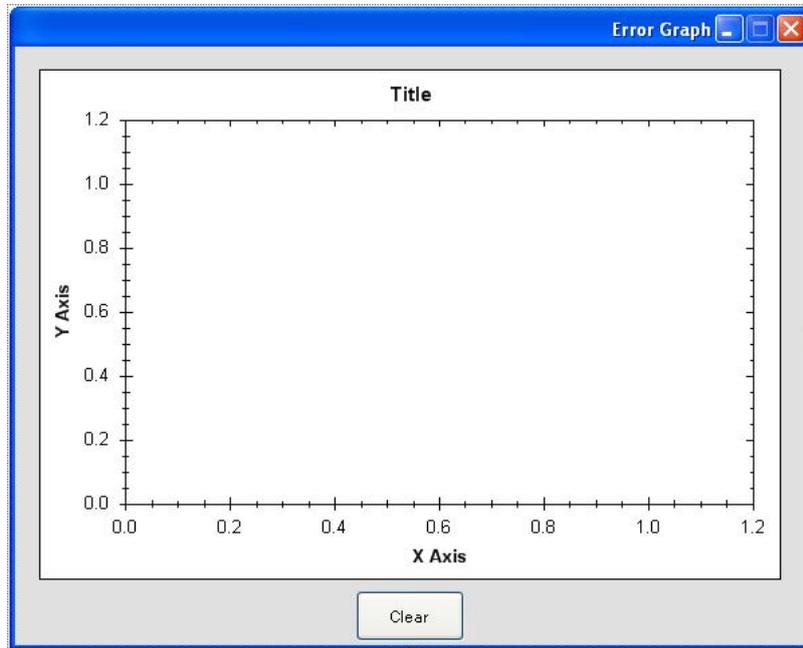


Figura 4.6: Controlo que permite visualizar a evolução do erro de treino.

Durante os testes realizados a esta ferramenta computacional foi encontrado um bug relacionado com este gráfico do erro. O gráfico de erro está a ser executado em paralelo com o treino da rede. Contudo, e devido às características do Microsoft Visual C#, os dois processos não estão a ser executados em simultâneo, como seria desejado. O tempo despendido a cada processo depende das características intrínsecas do computador onde o software está a correr fazendo com que se obtenha comportamentos distintos com o executar da ferramenta em máquinas distintas. A correcção desta característica indesejável é o próximo passo no nosso projecto assim como dotar o software de novas funcionalidades.

Bibliografia

- [1] http://en.wikipedia.org/wiki/Artificial_neural_network
- [2] A. Oliveira e J. Marques de Sá. *Ferramenta computacional para implementação de redes neuronais - a primeira versão*. Technical Report 2007-1, INEB-PSI, FEUP/DEEC, Rua Dr. Roberto Frias, 4200-645 Porto, Portugal, June 2007.