

Optimização da aplicação NNIG-Nets MLP
e desenvolvimento da aplicação NNIG-Nets RBF

Alexandra Oliveira

aao@fe.up.pt

Professor Joaquim Marques de Sá

December 2008

INEB - Instituto de Engenharia Biomédica

FEUP/DEEC, Rua Dr. Roberto Frias, 4200-645 PORTO

Conteúdo

1	Sobre as Ferramentas Computacionais	3
2	Acknowledgments	4
3	Principais fragilidades da ferramenta NNIG-Nets MLP	6
4	Aprefeiçoamento da interface gráfica do software	8
4.1	Activation Function Chooser	8
4.2	Backpropagation	9
4.3	Classification Matrix	10
4.4	Decision Border	11
4.5	Error Graph	11
4.6	Error Surface	12
4.7	Function Preview	12
4.8	Layer Properties	13
4.9	Janela principal do NNIG-Nets	13
4.10	Input Data	14
4.11	MLP	15
4.12	Output Preview	16
4.13	Graphical Network	16
4.14	Scatter Plot	17

5	Implementação do software NNIG-Nets RBF	18
5.1	Radial Basis Function Editor	19
5.2	Hybrid Training	20
5.3	Cluster plot	20
5.4	Algoritmos estruturantes	21
	Bibliography	22

Capítulo 1

Sobre as Ferramentas Computacionais

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([8]).

Os dois tipos de redes neuronais mais comuns são as *Multilayer Perceptron* e as *Radial Basis Functions*.

As redes do tipo *Multilayer Perceptron* foram exploradas em [1] e em [2].

As redes do tipo Radial Basis Function são um dos tipos de redes neuronais mais antigos e mais usados na comunidade científica. Elas apareceram associadas a diversas tarefas de classificação de dados como por exemplo clustering, interpolação funcional e modelos mistos [4]. Este tipo de rede, para além de ser muito eficiente, tem uma arquitectura simples: possuem apenas uma camada escondida e apenas a camada de saída tem pesos associados. Esta característica torna o treino desta rede neuronal fácil e rápido. As redes do tipo *Radial Basis Functions* podem ser usadas tanto para regressão como para classificação.

As redes do tipo *Multilayer Perceptron* encontram implementadas no software NNIG-Nets MLP cuja descrição pormenorizada se encontra em [1] e em [2]. As redes do tipo *Radial Basis Function* encontram-se implementadas numa ferramenta nova com título NNIG-Nets RBF.

Neste relatório descreve-se as principais alterações realizadas no software NNIG-Nets MLP assim como os aspectos mais importantes da ferramenta NNIG-Nets RBF.

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, Adriano Correia, Marco Sousa e Tânia Fontes 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 ferramenta

NNIG-Nets MLP

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. Alterar o control “Backpropagation” para permitir o acesso ao algoritmo de treino “Sequential Backpropagation”;
2. Identificar e corrigir os erros na controlo “Classification Matrix”;
3. Melhorar o funcionamento do controlo “Decision Border”;
4. Permitir ver o gráfico do erro de treino de duas corridas consecutivas;
5. Implementar algoritmos e desenvolver controlos que permitem a visualização da superfície de erro;
6. Substituir o controlo “Function Preview” por um mais eficiente;
7. Identificar e corrigir erros no controlo “Input Data”;
8. Reorganização do controlo principal da aplicação - o NNIG-NETs MLP;
9. Permitir aos utilizadores continuar com o treino das redes neuronais com número de iterações desejado e adaptar todos os controlos de visualização de resultados para esta nova funcionalidade;

10. Identificar e corrigir erros no controlo “Input Data”;
11. Alterar a estrutura do controlo “MLP” para que os pesos da rede fossem inicializados automaticamente com um valor aleatório entre -0,3 e 0,3;
12. Permitir que os dados de entradas se apresentem escritos em diferentes formatos;
13. Fazer a validação dos ficheiros que contêm dados relevantes ao funcionamento do software;

Capítulo 4

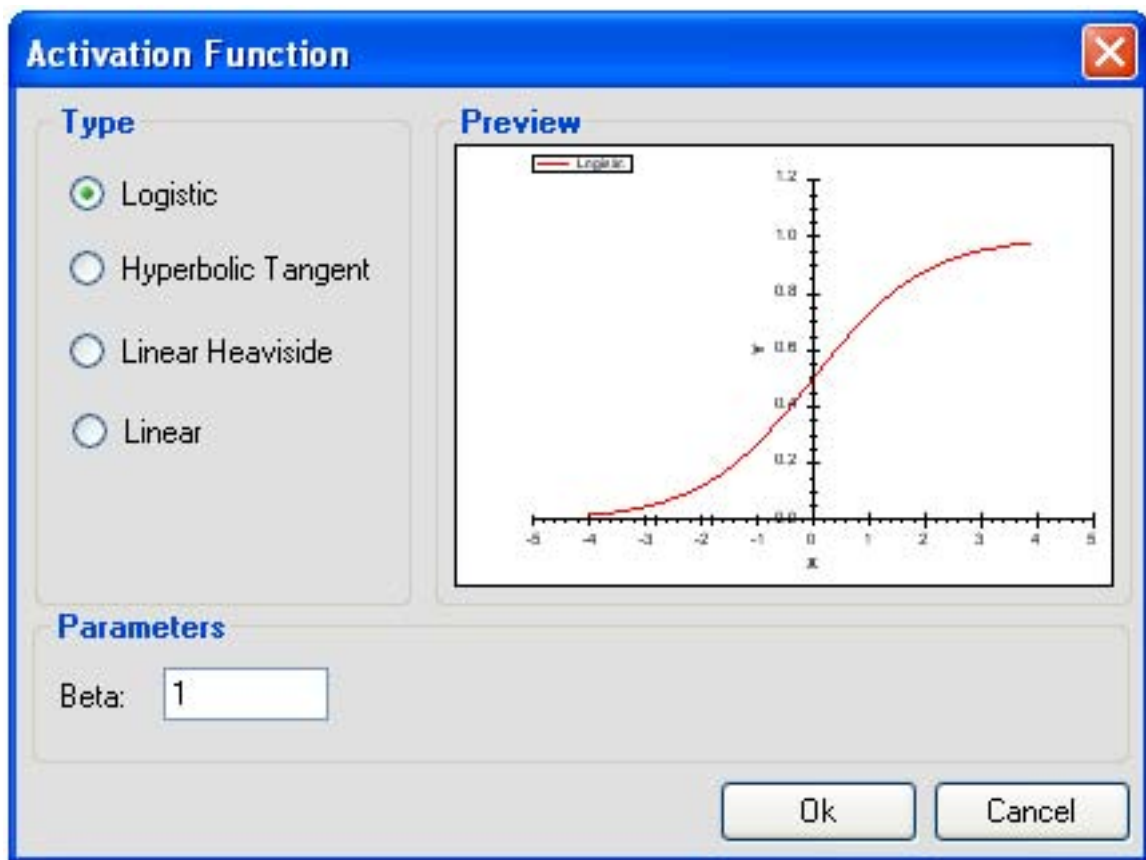
Aprefeioamento da interface gráfica do software

A necessidade de tornar os softwares NNIG-Nets MLP e NNIG-Nets RBF fáceis de usar por um utilizador comum foi premente. Desta forma a interface gráfica dos dois software sofreram várias alterações ao longo do ano de 2008. Em paralelo foram sendo identificados e corrigidos vários erros informáticos dos quais destaco os mais importantes. Todas os controlos que a versão de 2007 continha foram alteradas com a excepção da “all data preview”. Nas secções seguintes descrevem-se as principais alterações de cada um destes controlos.

4.1 Activation Function Chooser

Permite ao utilizador a escolha de funções de activação para os neurónios da rede neuronal artificial. Uma vez que a função linear que este controlo continha era, na realidade, uma função linear por bocados, foi adicionada uma função que representa toda a família de funções lineares. Uma vez que este controlo pode ser usado tanto em redes do tipo *MultiLayer Perceptron* como em *Radial Basis Functions*, e como estes dois tipos de redes usam funções de activação muito características, este controlo permite visualizar apenas as funções apropriadas a cada rede realizando uma pequena validação. Este controlo é comum a ambas as aplicações e encontra-se representado na figura seguinte:

O controlo que permite visualizar a função de activação foi substituído por um mais eficiente.



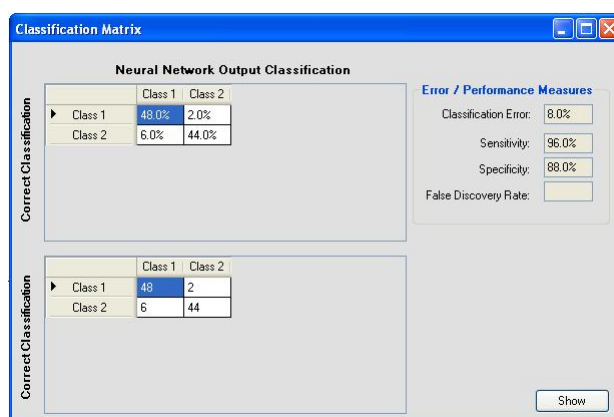
4.2 Backpropagation

Permite ao utilizador a escolha do algoritmo de treino a usar numa rede do tipo MultiLayer Perceptron. Visualmente o control Backpropagation não sofreu grandes alteração, apenas foram retiradas duas caixas de texto que permitiam visualizar dois resultados finais do algoritmo de treino. Contudo, passou a contar com a possibilidade de alternar entre os modos batch e sequencial, que são duas versões do mesmo algoritmo onde a actualização dos pesos é feita em momentos diferentes. Este controlo pertence apenas à aplicação NNIG-Nets MLP e encontra-se representado na figura seguinte:



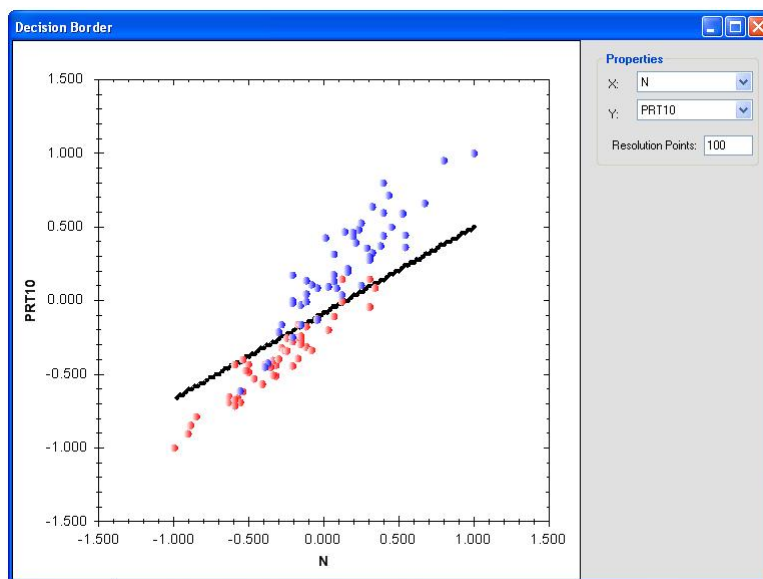
4.3 Classification Matrix

Permite a confrontação das classificações obtidas com as classificações reais. A matriz de classificação apresentava um erro na disposição de dados quando a tarefa era classificação de dados provenientes de mais de duas classes distintas. Este erro encontra-se corrigido. Foi adicionada uma segunda matriz que permite a visualização das distribuição da classificação dos dados em forma de percentagem. Foram ainda acrescentadas as seguintes medidas estatísticas: erro de classificação, sensibilidade, especificidade. Este controlo pertence a ambas as aplicações e encontra-se representado na figura seguinte:



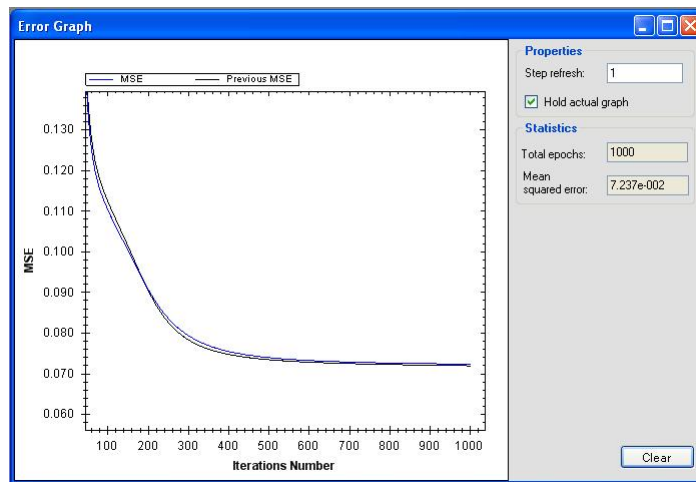
4.4 Decision Border

Permite a representação gráfica, em duas dimensões, das várias classes contidas nos dados assim como a representação gráfica da curva que a rede neuronal artificial determina como sendo a separação optima entre as classes. Este controlo apresentava um erro no desenho das curvas de separação das classes que já foi corrigido. Este controlo pertence a ambas as aplicações e encontra-se representado na figura seguinte:



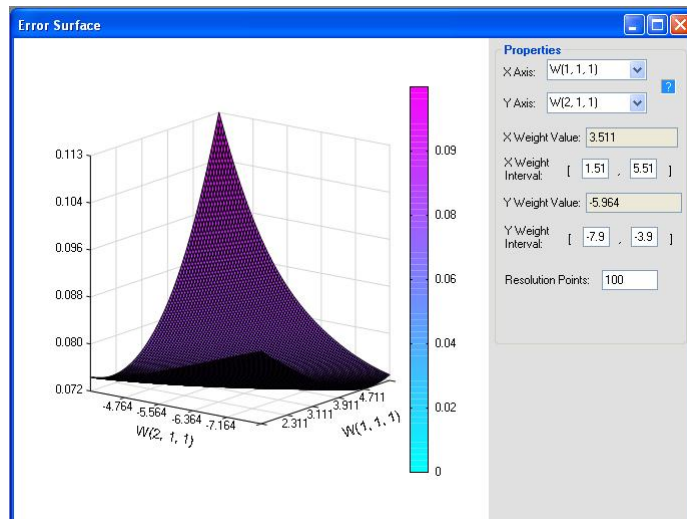
4.5 Error Graph

Permite a visualização da evolução do erro de treino ao longo dos diversos passos do seu processo. Possibilitou-se quer a visualização em tempo real de execução do treino quer após este processo estar terminado. Após o treino pode-se verificar o número de iterações realizadas assim como o erro final obtido. Este controlo permite ainda guardar a curva obtida na experiência realizada anteriormente para que se possa comparar as duas. É ainda possível visualizar a evolução do erro com a continuação de um algoritmo de treino iterativo. O Error Graph pertence à aplicação NNIG-Nets MLP uma vez que apenas este tipo de redes possui algoritmos de treino iterativos. Este gráfico encontra-se representado na figura seguinte dispondo o gráfico de duas experiências:



4.6 Error Surface

Este controlo foi implementado para que dados dois pesos da rede neuronal se possa visualizar em 3D a superfície de erro obtida pela rede neuronal. Este controlo pertence a ambas as aplicações e encontra-se representado na figura seguinte:

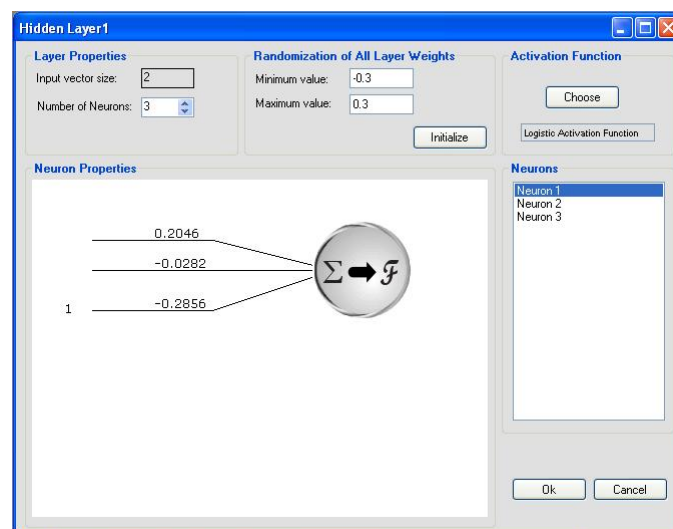


4.7 Function Preview

Este controlo foi substituído por um bastante mais eficiente: o zed graph [3].

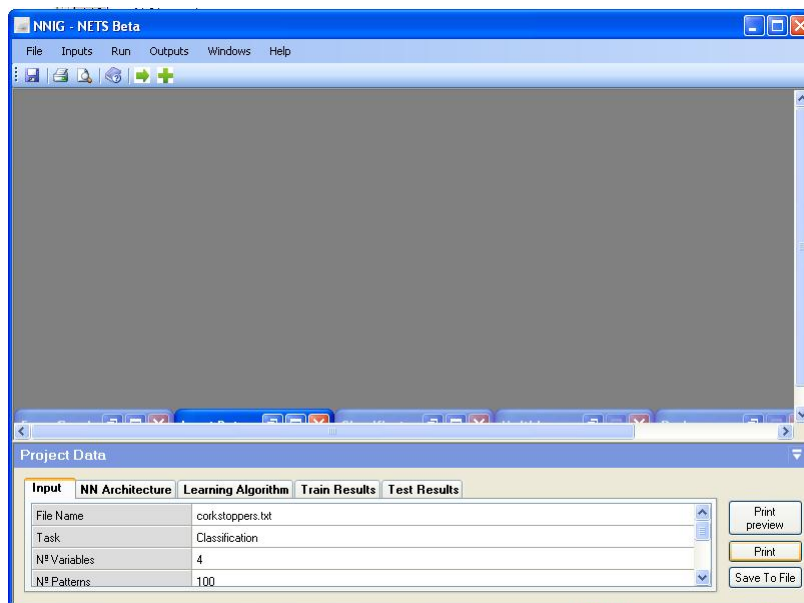
4.8 Layer Properties

Este controlo foi generalizado para que se possa editar o número de neurónios de uma determinada camada da rede neuronal do tipo Multilayer Perceptron. Para além do número de neurónios, permite escolher a função de activação para esta camada assim como os pesos que lhe estão associados. É ainda possível visualizar os pesos associados a cada neurónio. Este controlo pertence a ambas as aplicações e encontra-se representado na figura seguinte:



4.9 Janela principal do NNIG-Nets

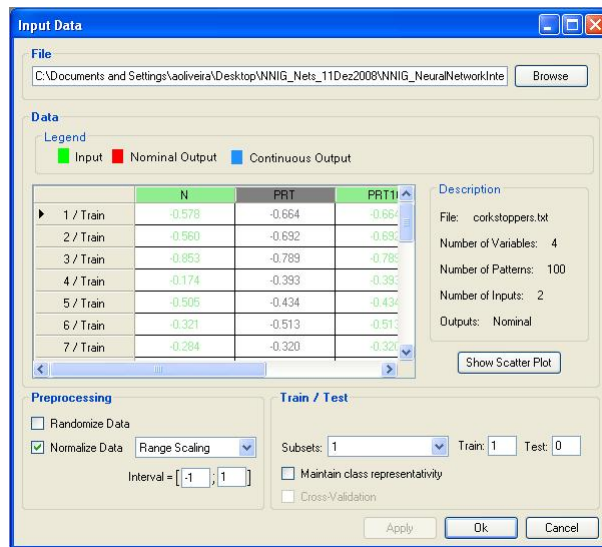
Esta janela foi re-organizada para permitir a generalização do software de modo a facilitar a incorporação de novos algoritmos de redes neuronais e algoritmos de treino correspondentes. Todos os controlos foram testados, assim como a sua interacção foi cuidadosamente desenhada devido às dependências entre eles. Além disso, foi ainda incorporada uma nova funcionalidade que consiste em continuar o treino de uma rede neuronal o número de iterações que deseje. Como o software está desenhado para que haja uma flexibilidade de escolhas, tanto nas opções sobre os dados de entrada, como nas opções sobre a arquitectura da rede neuronal e os parâmetros de algoritmo de treino, foi criado um painel nesta janela que vai registando todas as opções do utilizador. A qualquer momento estas opções podem ser guardadas ou impressas.



4.10 Input Data

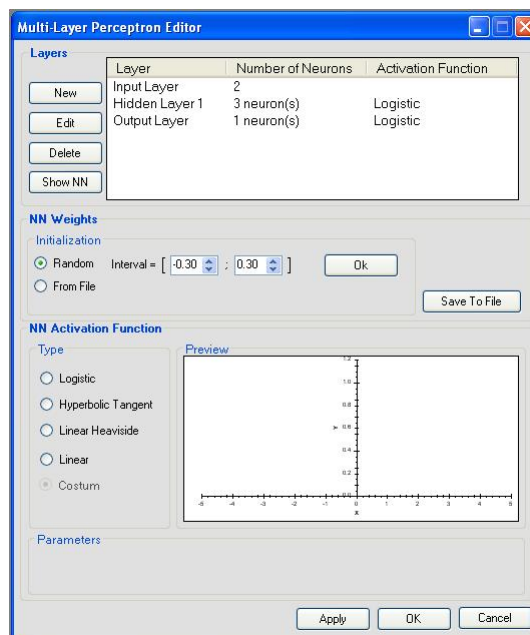
Desde o ano de 2007 este controlo foi o que sofreu mais alterações. No final do ano passado este controlo apresentava várias fragilidades que geravam erros no software. Este controlo não fazia a referência correcta às variáveis contidas nos ficheiro em formato .txt e sempre que o número de variáveis de um ficheiro era alterada, o software dava erro. Estas características foram devidamente corrigidas. O software permite ainda que os dados de entrada (em formato .txt) sejam apresentados nos formatos mais usados pela comunidade científica, ou seja, as variáveis podem estar separadas por ponto e vírgula ou por tab.

Depois dos dados introduzidos na aplicação, é possível fazer a divisão dos dados em dois conjuntos disjuntos, um para treino da rede e outro para teste. Pode-se ainda ordenar de forma aleatória os dados e normalizá-los. Todas estas acções são reversíveis e a qualquer momento o utilizador pode voltar a ter os dados originais. Dado um ficheiro, o software recolhe automaticamente informações sobre o número de características e sobre o número de padrões. Este controlo pertence a ambas as aplicações e encontra-se representado na figura seguinte:



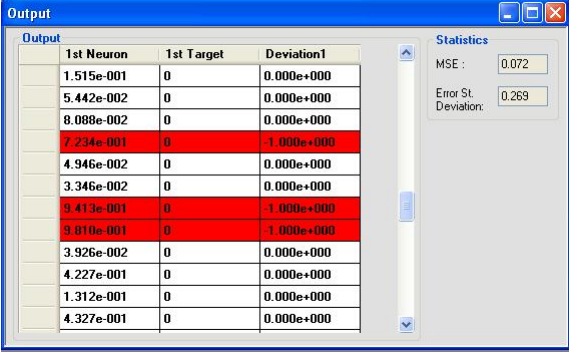
4.11 MLP

O controlo “MLP” permite definir a arquitectura da rede neuronal. Desde o ano passado que este faz a inicialização dos pesos da rede neuronal de forma automática aleatória e no intervalo de -0,3 a 0,3. Permite ainda atribuir pesos iniciais que se encontram registados num ficheiro fazendo a devida validação do formato do ficheiro. Os pesos podem ainda ser guardados num ficheiro .txt. Este controlo pertence à aplicação NNIG-Nets MLP e encontra-se representado na figura seguinte:



4.12 Output Preview

Este controlo foi totalmente desenvolvido durante o ano de 2008 e permite visualizar as saídas das redes neuronais em confronto com os exemplos usados para aprendizagem. Apresenta ainda as diferenças entre as saídas e os exemplos. No caso de a tarefa ser classificação de dados, o controlo assinala os padrões mal classificados. Apresenta o erro médio obtido na tarefa assim como o respectivo devido desvio padrão. Este controlo pertence a ambas as aplicações e encontra-se representado na figura seguinte:



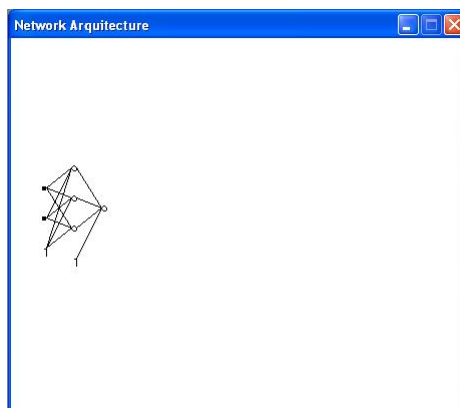
The screenshot shows a window titled "Output" with a table of data and a statistics panel on the right. The table has three columns: "1st Neuron", "1st Target", and "Deviation1". The "Deviation1" column has red background for rows where the deviation is -1.000e+000. The statistics panel shows MSE: 0.072 and Error St. Deviation: 0.269.

1st Neuron	1st Target	Deviation1
1.515e-001	0	0.000e+000
5.442e-002	0	0.000e+000
8.088e-002	0	0.000e+000
7.234e-001	0	-1.000e+000
4.946e-002	0	0.000e+000
3.346e-002	0	0.000e+000
9.413e-001	0	-1.000e+000
9.010e-001	0	-1.000e+000
3.926e-002	0	0.000e+000
4.227e-001	0	0.000e+000
1.312e-001	0	0.000e+000
4.327e-001	0	0.000e+000

Statistics:
MSE : 0.072
Error St. Deviation: 0.269

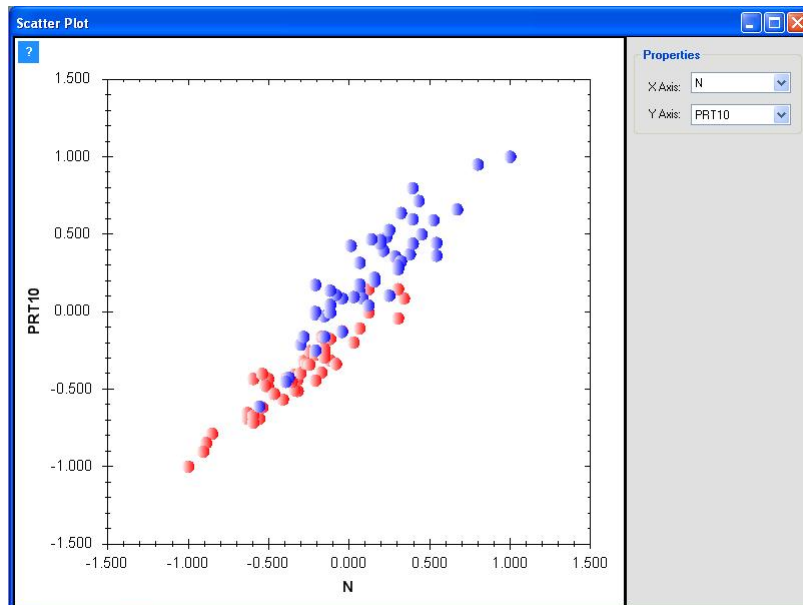
4.13 Graphical Network

Permite representar graficamente a arquitectura de uma rede neuronal. Este controlo pertence a ambas as aplicações e encontra-se representado na figura seguinte:



4.14 Scatter Plot

Permite a visualização dos dados de entrada em 2D. No caso da tarefa da rede neuronal ser classificação de dados, este controlo permite a diferenciação das classes. Este controlo pertence a ambas as aplicações e encontra-se representado na figura seguinte:



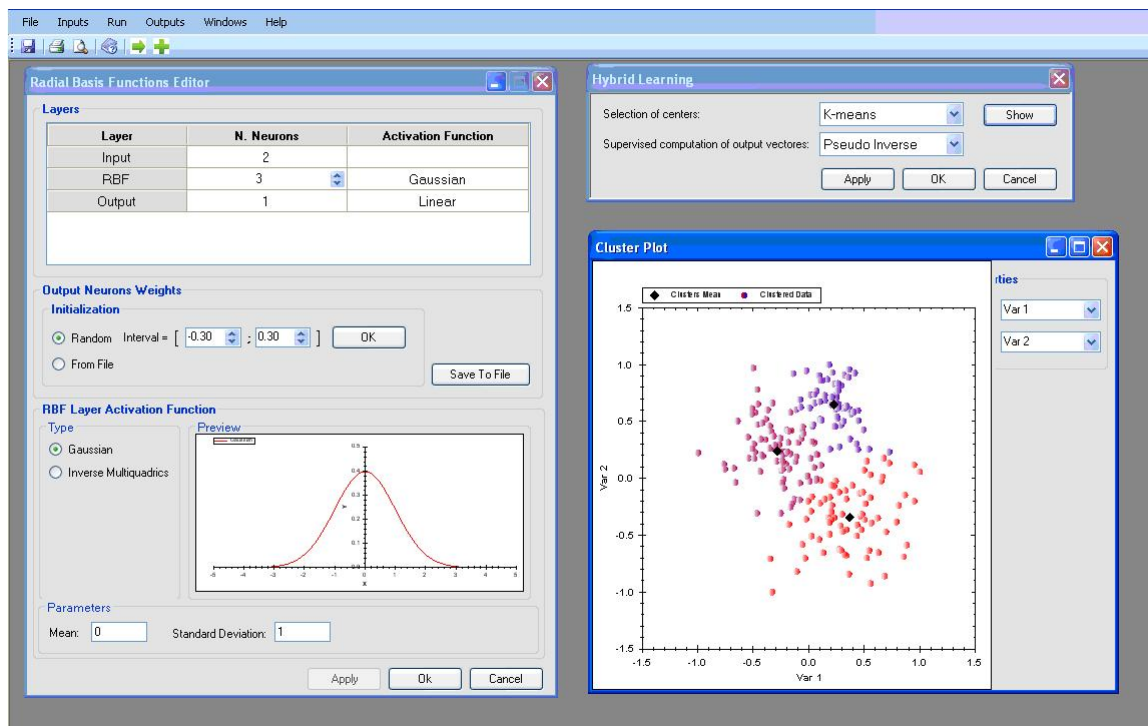
Capítulo 5

Implementação do software NNIG-Nets RBF

Foi desenvolvido uma nova aplicação com o nome NNIG-Nets RBF para a integração dos algoritmos referentes às redes do tipo *Radial Basis Function*.

Os controlos de entrada e saída de dados desta aplicação que são comuns com à aplicação NNIG-Nets MLP encontram-se identificados e descritos no capítulo anterior enquanto que os controlos característicos desta aplicação serão descritos nas secções seguintes.

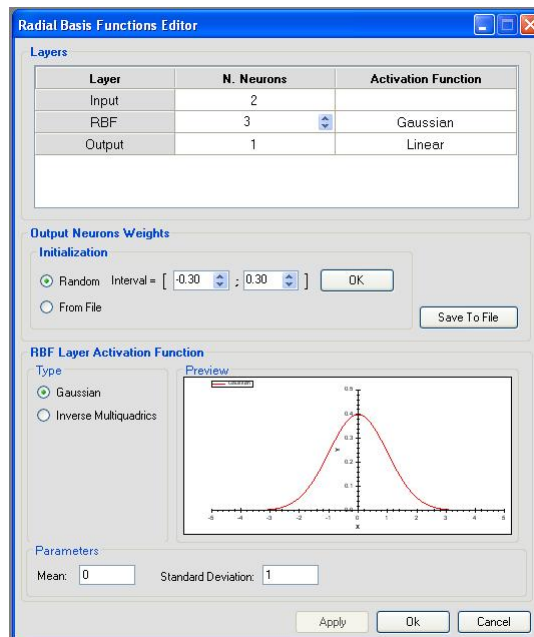
As principais janelas desta aplicação encontram-se representadas na figura seguinte:



5.1 Radial Basis Function Editor

Permite a visualização e alteração da arquitectura das redes do tipo RBF (*Radial Basis Function*). Este tipo de redes tem uma arquitectura fixa sendo constituída por uma camada escondida com um número arbitrário de neurónios e por uma camada de saída com função de activação linear. A camada escondida pode ter um dos seguintes tipos de funções de activação: gaussiana ou função inversa da função multiquadrada. O número de neurónios desta camada determina o número de centros de clusters que a rede vai usar e que são determinados pelo algoritmo K-means ou indicados pelo utilizador (ambos encontram-se no controlo “Hybrid Training”). O “Radial Basis Function Editor” permite que se visualize e altere os pesos da camada de saída. A função de activação desta camada é a função identidade e não pode ser alterada.

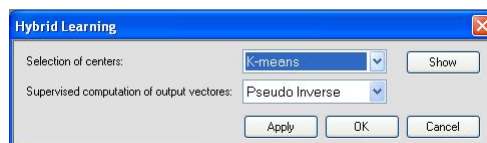
O controlo “Radial Basis Function Editor” encontra-se representado na figura seguinte:



5.2 Hybrid Training

As redes do tipo RBF têm, na aplicação NNIG-NETS RBF, um algoritmo de treino híbrido, ou seja, é processado em dois momentos diferentes: -um algoritmo não supervisionado para a escolha dos centros - k-means ou escolhidos pelo utilizador; - um algoritmo supervisionado para o ajuste dos pesos - a determinação da matriz pseudo-inversa da matriz design. A partir do momento que se escolhe o método k-means para a determinação dos centros das redes do tipo *Radial Basis Function* pode-se visualizar estes centros no controlo “Cluster plot”.

Este controlo encontra-se representado na figura seguinte:

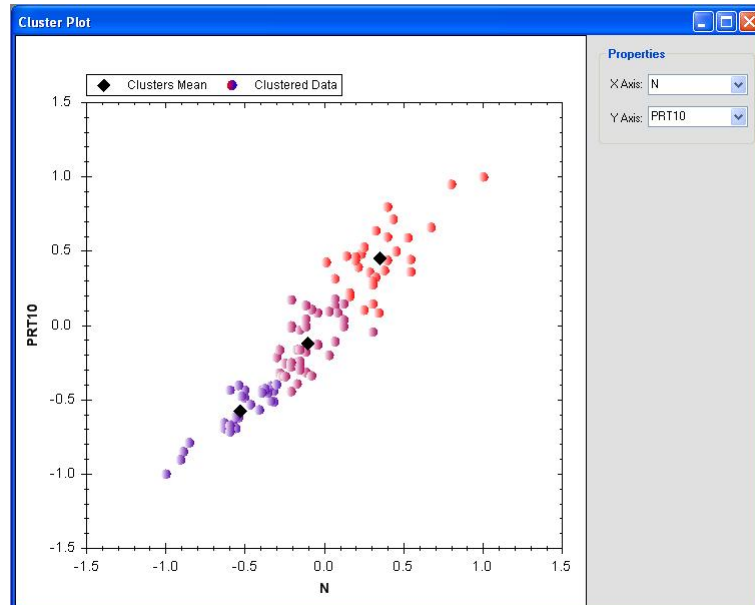


5.3 Cluster plot

Permite a visualização dos clusters determinados pelo algoritmo k-means assim como o seu centro. Cada cluster encontra-se identificado por uma cor distinta. O número de clusters é determinado pelo número de neurónios contidos na camada escondida da arquitectura da rede RBF.

Na figura seguinte encontra-se representado o resultado de um processo de clustering pelo pro-

cesso K-means para uma rede com três neurónios na camada escondida:



5.4 Algoritmos estruturantes

Para criar uma rede do tipo RBF foi necessário criar classes específicas com todas as funcionalidades e características deste tipo de redes. Implementou-se também algoritmos para inversão de matrizes bem como o algoritmo de k-means. Estes algoritmos foram criados com base em [7][6] [5].

Bibliografia

- [1] 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.
- [2] A. Oliveira e J. Marques de Sá. *Ferramenta computacional para implementação de redes neuronais optimização das capacidades gráficas*. Technical Report 2007-1, INEB-PSI, FEUP/DEEC, Rua Dr. Roberto Frias, 4200-645 Porto, Portugal, December 2007.
- [3] http://zedgraph.org/wiki/index.php?title=Main_Page
- [4] Bors Adrian, *Introduction of the Radial Basis Function (RBF) networks*
(<http://axiom.anu.edu.au/~daa/courses/GSAC6017/rbf.pdf>)
- [5] Press W. et all *Numerical Recipes in C*, Cambridge University Press,1992
(<http://www.nrbook.com/nr3/>)
- [6] <http://www.alglib.net/matrixops/general/inv.php>
- [7] http://cs.smu.ca/~r_zhang/code/kmeans.c
- [8] http://en.wikipedia.org/wiki/Artificial_neural_network