

UNIVERSIDADE ESTADUAL DE MONTES CLAROS
CENTRO DE CIÊNCIAS EXATAS E TECNOLÓGICAS
DEPARTAMENTO DE CIÊNCIAS DA COMPUTAÇÃO
ESPECIALIZAÇÃO EM ENGENHARIA DE SISTEMAS
DISCIPLINA: REDES NEURAIIS
PROFESSOR: MARCOS FLÁVIO DÂNGELO
ALUNOS: DIOGO DE SOUZA MENDES
LUCIANA XAVIER BRANT

Relatório de uma Aplicação de Redes Neurais

Março – 2011

1 - INTRODUÇÃO

Pretendemos desenvolver uma aplicação de uma Rede Neural Artificial para prever o consumo de água em uma residência. A rede utiliza os valores de medição dos quatro últimos meses para prever o valor do próximo mês. Utilizaremos valores simples para facilitar o entendimento e a visualização dos resultados

Para implementação da aplicação utilizamos as redes neurais do Toolbox do software Matlab utilizado linhas de comando e uma ferramenta de interface gráfica chamada NNTool.

2 - DEFINIÇÃO

Redes Neurais são técnicas computacionais que apresentam um modelo matemático inspirado na estrutura neuronal de organismos inteligentes e que adquirem conhecimento através da experiência.

Baseado no funcionamento do cérebro humano, ou no procedimento como o cérebro aprende, as redes neurais artificiais são capazes de simular conexões sinápticas. Cada neurônio do cérebro humano tem a capacidade de criar até 10.000 sinapses com outros neurônios.

O neurônio biológico é composto pelo corpo do neurônio (ou soma), pelos dendritos e axônios; o corpo do neurônio é responsável por coletar e combinar informações vindas de outros neurônios, os dendritos recebem estímulos e o axônio transmite esses estímulos. Um neurônio recebe sinais através dos dendritos, onde é ponderado e enviado ao axônio, um sinal pode ser ampliado ou reduzido, por que a cada passagem por um neurônio está associado um peso que é multiplicado ao sinal. (Ludwig Jr; COSTA, 2007)

A sinapse é o processo de ligação entre o axônio e o dendrito; para que o sinal entre no próximo neurônio, deve ocorrer sinapse. Mas para isso, esse sinal deve ser superior a certo limite, que segundo Ludwig Jr; COSTA (2007, p 10) deve ser superior a 50 mV (limiar de disparo), caso não for é bloqueado.

O neurônio artificial trabalha de forma similar ao neurônio biológico, recebendo sinais de entrada e distribuindo para a saída ou para outros neurônios da camada seguinte. O sinal de entrada é multiplicado pelos pesos sinápticos, gerando um somatório de todas as entradas ponderadas, esse somatório é considerado a função de ativação para que possa transmitir o sinal para outros neurônios ou para a saída.

O Matlab (Matrix Laboratory) é um sistema interativo e uma linguagem de programação para a computação técnica e científica em geral, ele é baseado em matrizes. O Matlab possui uma família de aplicativos específicos (*toolboxes*), que são coleções de funções usadas para resolver determinados problemas tais como: otimização, manipulação algébrica, redes neurais, processamento de sinais, simulação de sistemas dinâmicos, entre outros.

O Matlab trabalha com duas janelas: uma janela de comandos que é usada para a entrada dos comandos, dados, e para mostrar os resultados na tela, e outra janela gráfica onde são gerados os gráficos.

Para utilizar as redes neurais no Toolbox do Matlab pode ocorrer de duas formas: linha de comando ou interface gráfica, chamada NNTOOL.

3 – IMPLEMENTAÇÃO

Para a criação da rede neural devem-se seguir os seguintes passos: definir os padrões (entrada e saída), inicializar a rede, definir os parâmetros de treinamento, treinar a rede e testar a rede.

Faremos a implementação utilizando linha de comando e a ferramenta de interface gráfica NNTool do Matlab versão 7.5.0.342 (R2007b). Utilizaremos dados fictícios de medidas de água de uma residência em diversos meses do ano com objetivo de treinar a rede e possibilitar a previsão da medida de um mês baseado nas medidas dos quatro meses anteriores.

Os dados utilizados como parâmetros de treinamento podem ser visualizados na tabela 1.

Mês	FEV	JAN	DEZ	NOV	OUT	SET	AGO	JUL
Medida	292	270	287	272	261	316	294	269

Tabela 1 – Parâmetros de treinamento

3.1 Entendendo o funcionamento da Linha de Comando

Seguindo esses passos a partir da linha de comando do Matlab.

Digita os valores de entrada e saída - definindo os Padrões

```
X = [ 0 0 11; 0 1 0 1]  
Y = [0 1 1 0]
```

Definindo a estrutura da rede

```
net = newff( [min(P)' max(P)],[N_hidden 1],{'tansig' 'logsig'},'traingd');
```

newff_ cria a estrutura de uma rede neuronal directa com retropropagação (feed-forward backpropagation network)

Mínimo e máximo da matriz de entrada_ limites dos padrões de entrada
N_hidden 1_ número de neurônios de cada camada

'tansig' ou 'logsig' _ função de ativação de cada camada
'traingd' _ algoritmo de treinamento

Definindo parâmetros de treinamento

net.trainParam.epochs = 100;	Número de epochs
net.trainParam.goal = 1e-8;	Erro final desejado
net.trainParam.lr = 0.01;	Taxa de aprendizado
net.trainParam.show = 25;	Atualização da tela (epochs)
net.trainParam.mc = 0.9;	Taxa de momentum
net.trainParam.lr_inc = 1.05;	Taxa de incremento da l.r.
net.trainParam.lr_dec = 0.7;	Taxa de decremento da l.r.
net.trainParam.max_perf_inc = 1.04;	Incremento máximo do erro

Treinando a Rede Neural

```
net = train(net, P, T);
```

Testando a Rede Neural

```
C = sim(net, P);
```

3.2 Implementação utilizando Linha de Comandos

%Definindo as entradas que servirão pra o treinamento da rede

```
P = [270 287 272 261;  
     287 272 261 316;  
     272 261 316 294;  
     261 316 294 269];
```

%Definindo os targets(legendas) que servirão para o treinamento da rede

```
T = [292 270 287 272];
```

%Instanciando a rede com 16 neurônios na camada oculta e 1 neurônio na camada de saída

```
net = newff(minmax(P),[16 1],{'tansig' 'tansig'});
```

%Comando par realizar o treinando a rede

```
net = train(net,P,T);
```

%Definindo valores de entradas para a rede calcular a previsão de saída

```
S=[80; 60; 40; 20];
```

%Fazendo a rede simular com os valores das entradas dadas no passo anterior

```
Y = sim(net,S);
```

%Exibindo o resultado

Y

4 IMPLEMENTAÇÃO UTILIZANDO A FERRAMENTA NNTOOL

O Matlab também possibilita a criação Redes Neurais Artificiais de forma simples e intuitiva utilizando a ferramenta NNTool. Em interface gráfica é possível definir os padrões de entrada, saída e modelar o tipo de rede utilizando os diversos recursos desta ferramenta.

4.1 – Tela inicial do NNTool

Para executar o aplicativo basta digitar o comando “**nntool**” na linha de comandos do Matlab. Então surgirá a janela inicial demonstrada na Figura 1.

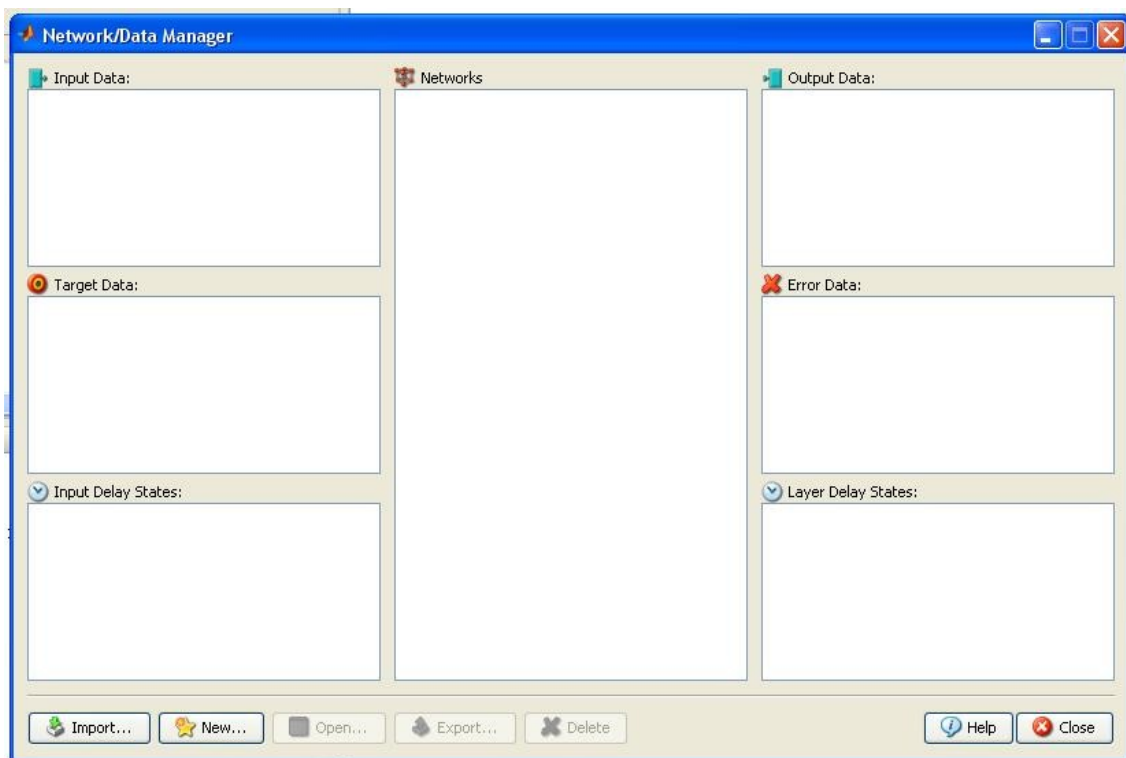


Figura 1 - Tela inicial do NNTool

Na janela inicial temos vários quadros onde podemos ver e alterar os conjuntos de dados de entrada, saída, padrões de saída e erros. Temos também botões para importar ou exportar dados, criar, abrir e excluir uma rede neural.

4.2 – Criando uma nova Rede Neural Artificial.

Pra criar uma nova RNA ou um novo conjunto de dados para entrada (inputs) ou padrão de saída (targets) devemos pressionar o botão “**New**” da tela inicial. Em seguida será mostrado um nova janela de título “*Create Network or Data*” demonstrada na Figura 2. Nesta Janela é possível alternar as abas para criar Redes Neurais ou Dados. É importante ressaltar que para criar uma nova RNA devemos passar os valores de entrada

e se for do tipo supervisionada devemos passar os padrões de saída também. Sendo assim, não é possível criar uma nova RNA sem criar primeiro os padrões de dados requisitados por ela.

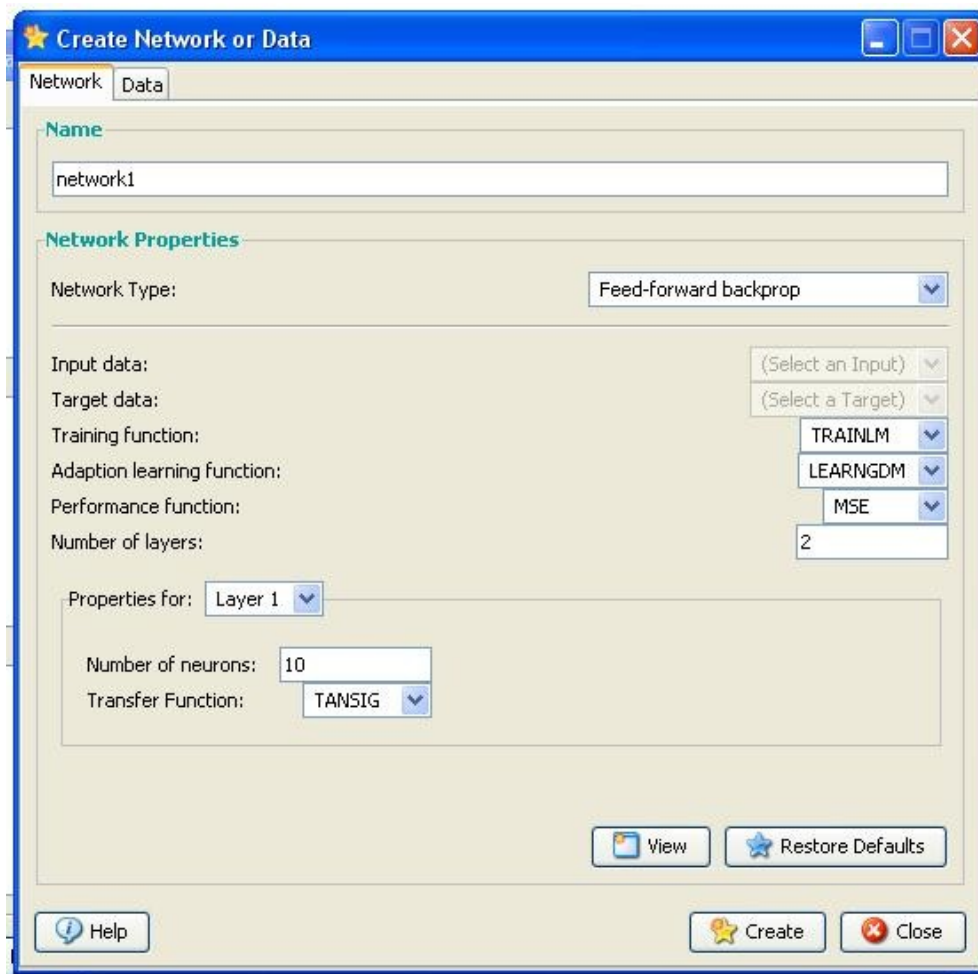


Figura 2 – Janela para criar uma nova RNA ou Dados

4.2.1 Criando os Conjuntos de Dados necessários para a RNA

Na aba **“Data”** podemos inserir os diversos parâmetros para a rede. Os primeiros dados a serem inseridos serão os conjuntos de dados da entrada que representam quatro intervalos de quatro meses cada. Para este grupo de informações daremos o nome **“P”**. Na figura 3 podemos ver o nome da entrada e os valores da mesma.

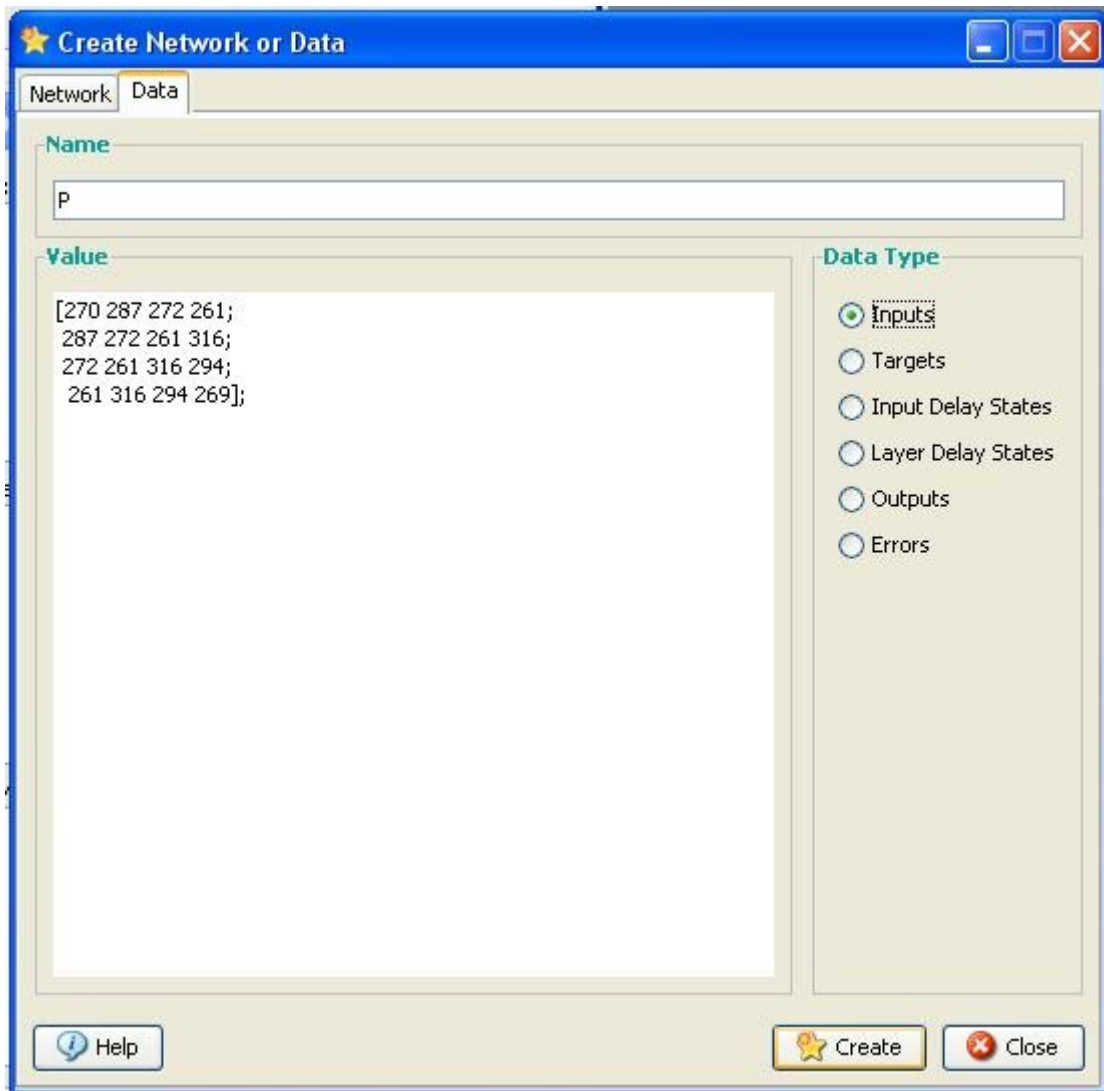


Figura 3 – Valores de Entrada

Na entrada "**P**" temos quatro seqüência de quatro entradas e para cada uma destas seqüências temos um valor de saída especificadas no elemento "**T**" do tipo *target*. A figura 4 mostra a criação dos *targets*(legendas) para cada grupo de entradas.

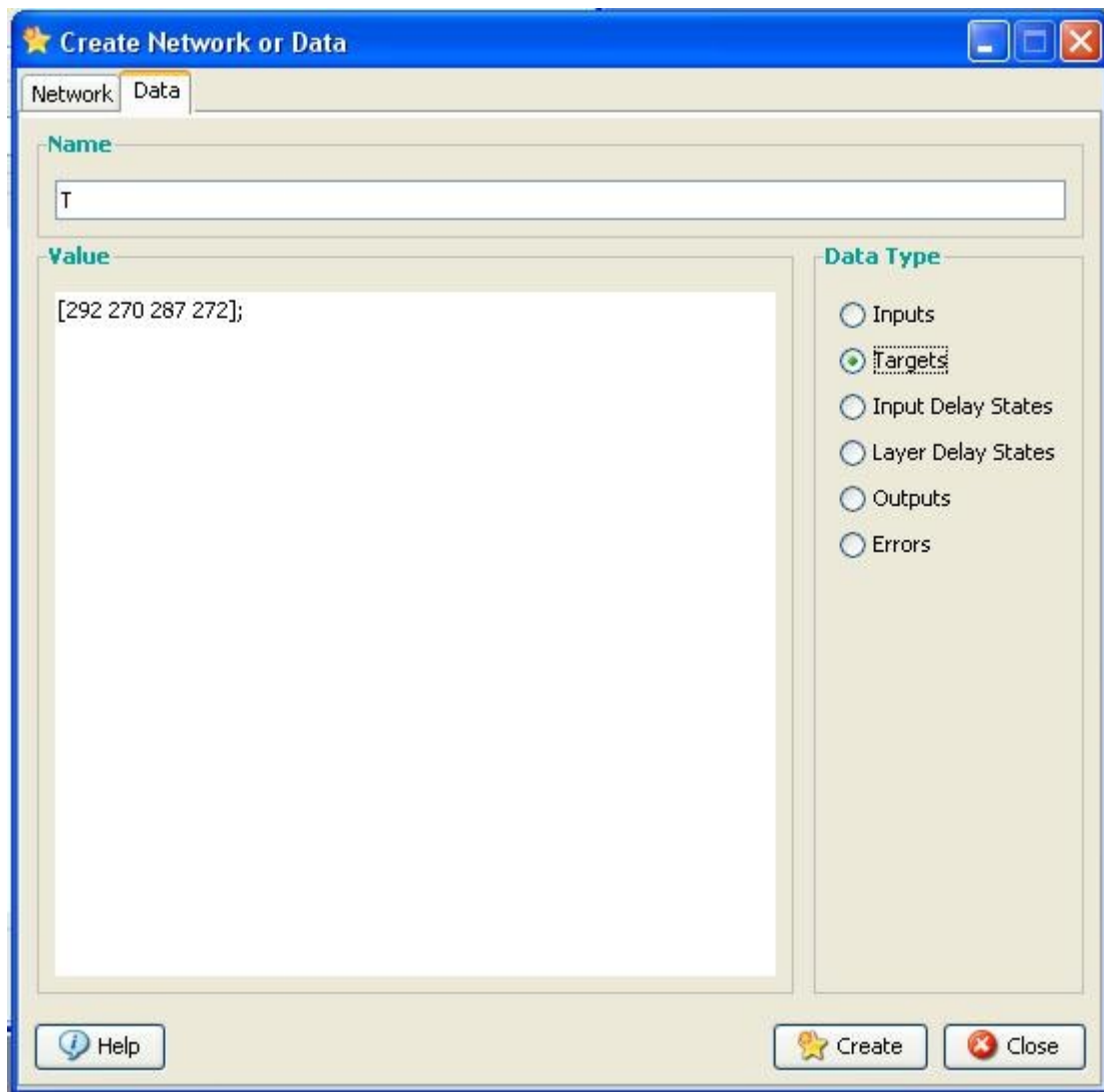


Figura 4 – Padrões de Saída (*Targets*)

O Próximo passo será inserir valores de entrada(*inputs*) para realizar simulações. Vamos inserir valores de quatro meses consecutivos (Outubro, Setembro, Agosto e Julho) e armazenar na variável **S1**. A Figura 5 demonstra a entrada dos valores para realizar a simulação.

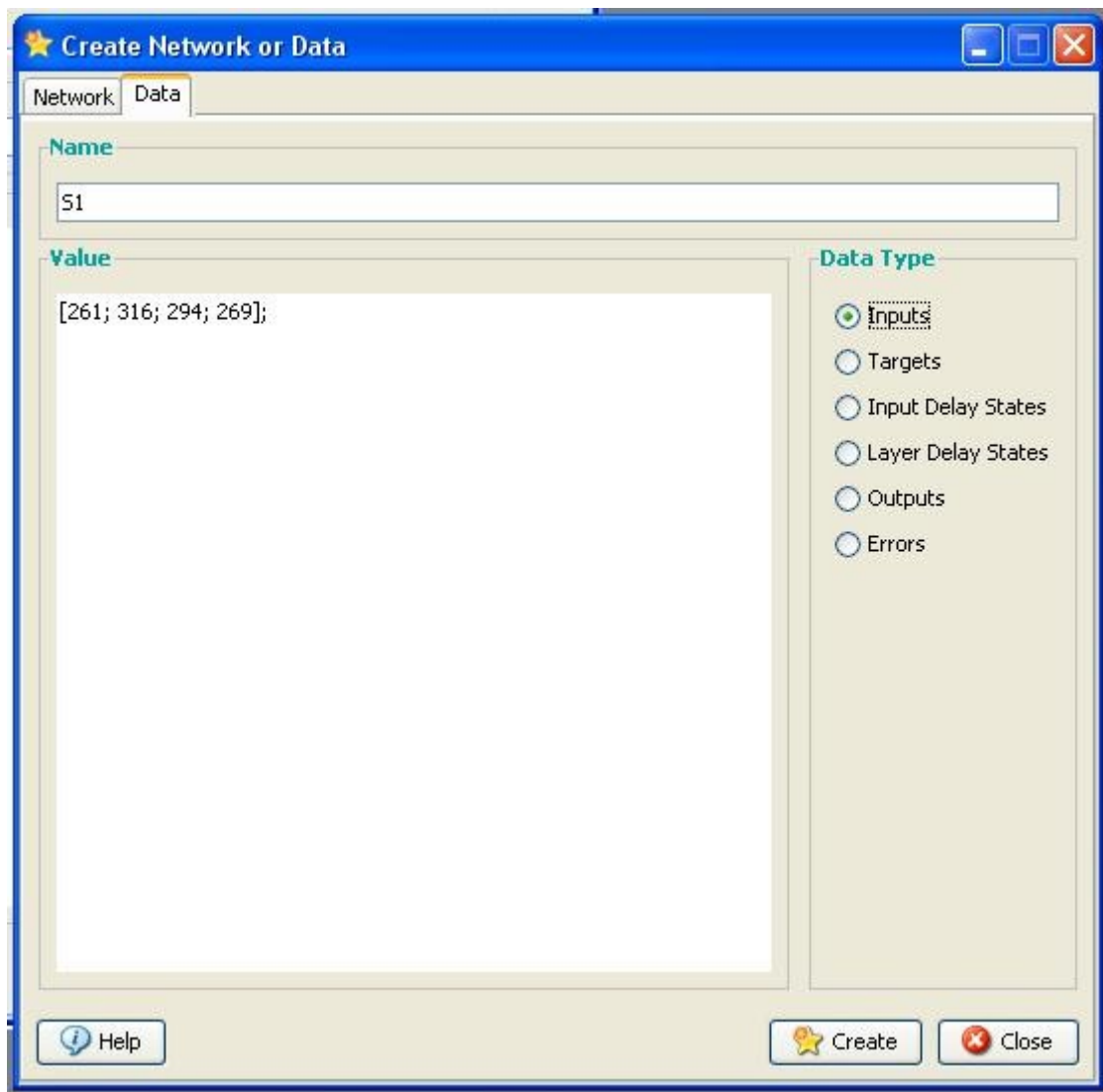


Figura 5 – Entrada para Simulação

4.2.2 Criando a nova RNA

Já temos os dados para a rede trabalhar agora o próximo passo é criar uma nova Rede na aba *Network*. Para criar uma nova rede devemos dar um nome a ela. Escolher o tipo de rede. Escolher os dados de entrada e legendas para o treinamento. O tipo de função para o treinamento, a quantidade de camadas e a função de transferência para cada camada. A figura 6 mostra a configuração da nossa rede. Com os seguintes valores:

Nome (*Name*): Consumo_de_Agua_01
 Tipo de Rede (*Network Type*): Feed Forward backprop
 Padrão de Entrada (*Input Data*): P
 Legendas (*Target Data*): T
 Função de Treinamento (*Training Function*): Trainlm
 Função de Adaptação de treinamento (*Adaption Learning Funcion:*
Função de Adaptação de Treinamento): Learngdm
 Função de Performance (*Performance Function*): MSE

Numero de Camadas (Number of Layer): 2
Propriedade das Camadas (Propreties for)
Camada1(Layer1) – Números de Neurônios:16
Função de Transferência: Tansig
Camada2(Layer2) – Numero de Neurônios: 1
Função de Transferência: Tansig

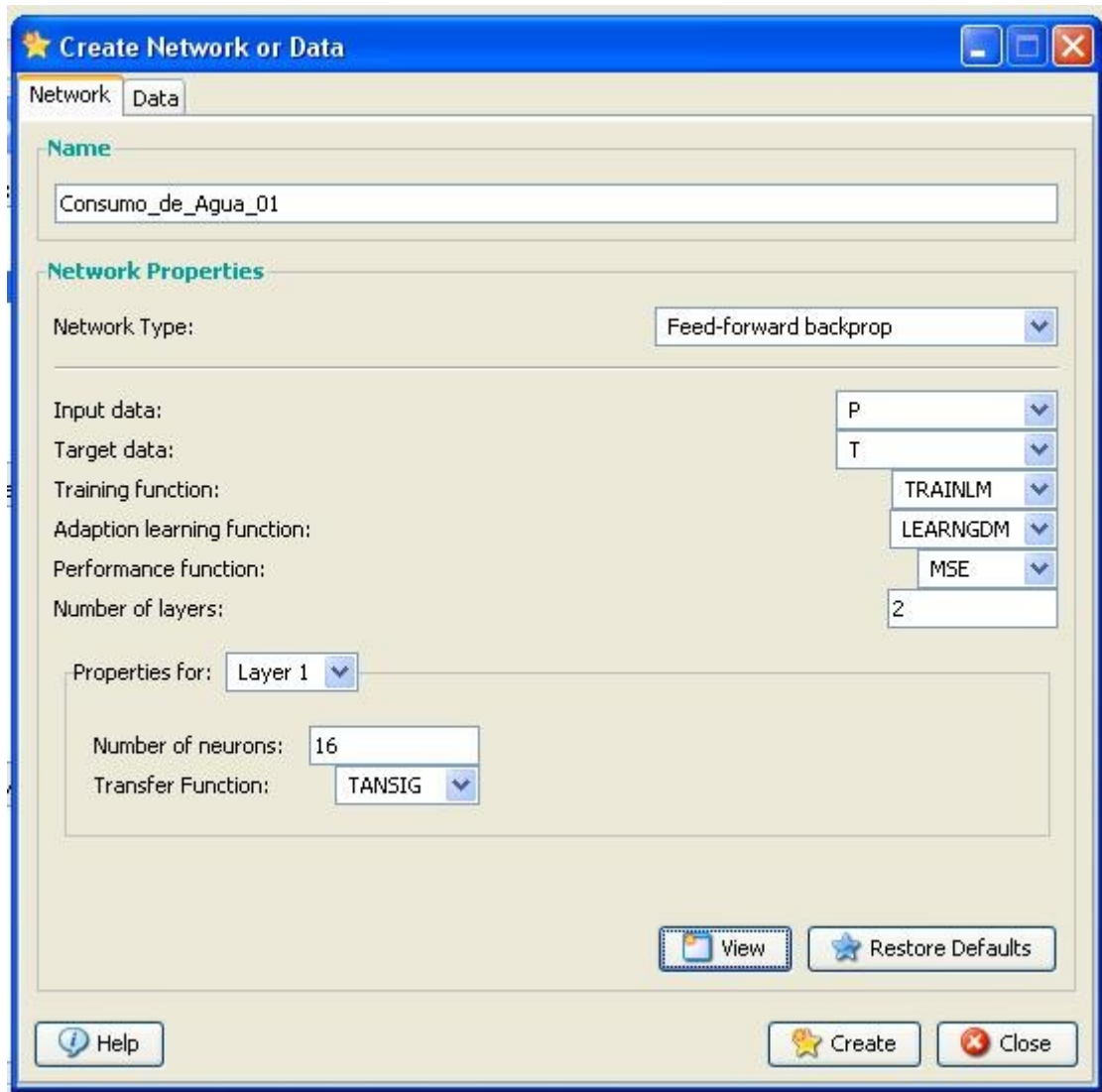


Figura 6 – Criando uma nova Rede

Agora podemos observar na figura 7 todos os dados de entrada, legendas e redes criadas.

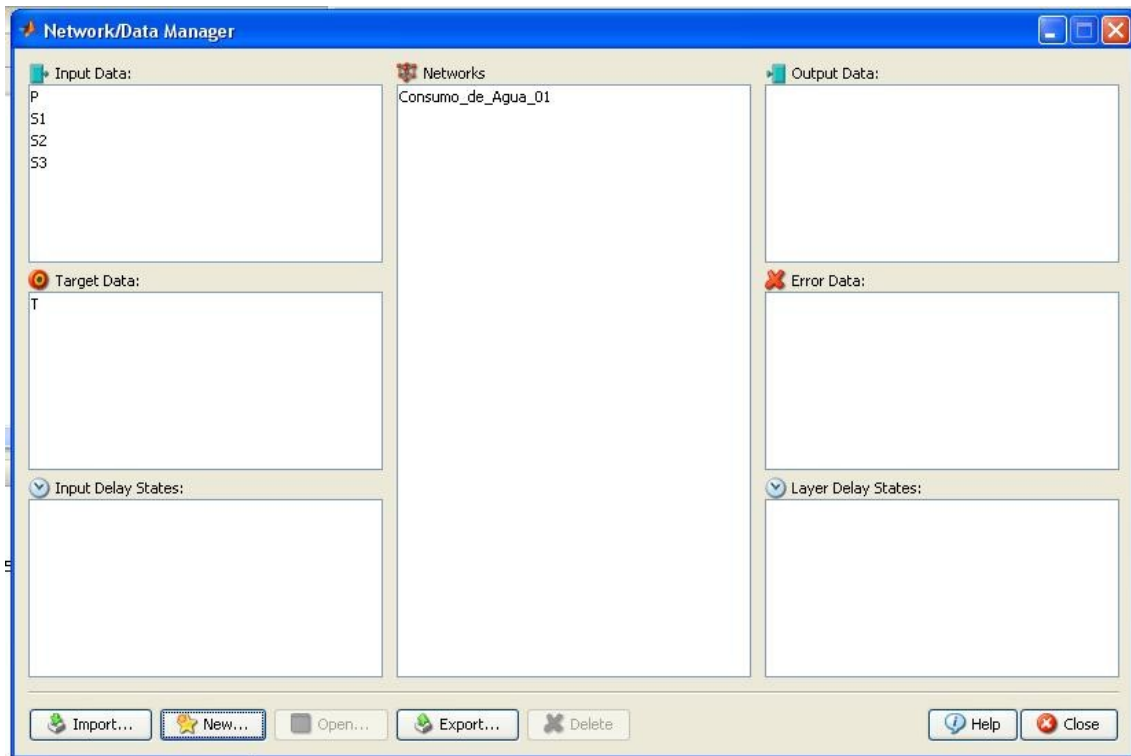


Figura 7 – Tela inicial do NNTool com dados e rede criados.

4.2.3 Treinamento da Rede

Com a rede pronta já podemos treiná-la e realizar os teste de simulação. Para acessar a rede podemos clicar duas vezes no seu nome ou clicar uma vez e pressionar o botão *open* na parte inferior da janela do NNTool. Após abrir já podemos visualizar o formato de rede criada onde é possível ver a quantidade de entradas, camadas e neurônios em cada camada. Na figura 8 podemos visualizar o diagrama da nossa RNA.

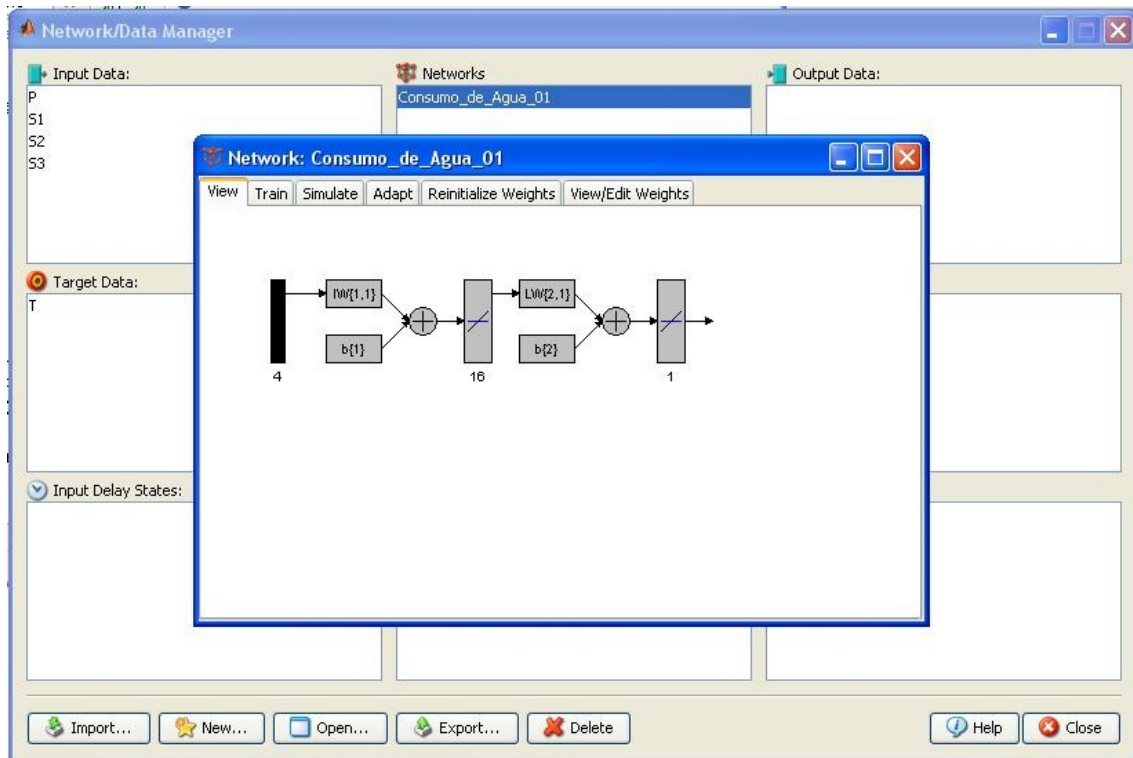


Figura 8 – Diagrama da Rede

O próximo passo é treinar a rede. Para isso devemos clicar na aba *Train* da rede criada. Nesta aba devemos passar as informações para treinamento e definir os nomes das variáveis de saída. Para nossa rede vamos inserir os valores de treinamento criados anteriormente armazenados nas variáveis de entrada **P** e legendas **T**. Na figura 9 podemos a tela de configuração dos dados para treinamento.

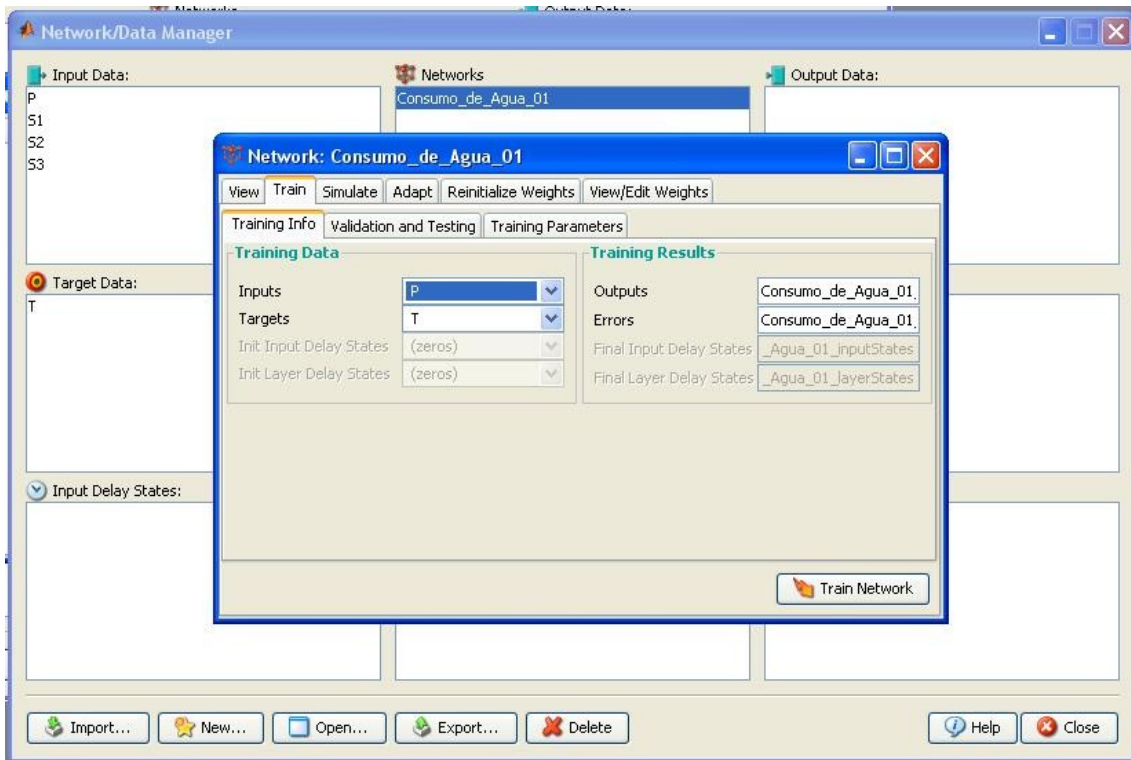


Figura 9 – Configuração dos dados para o treinamento.

Podemos modificar parâmetros avançados de treinamento na sub-aba *Training Parameters*(Parâmetros de Treinamento).

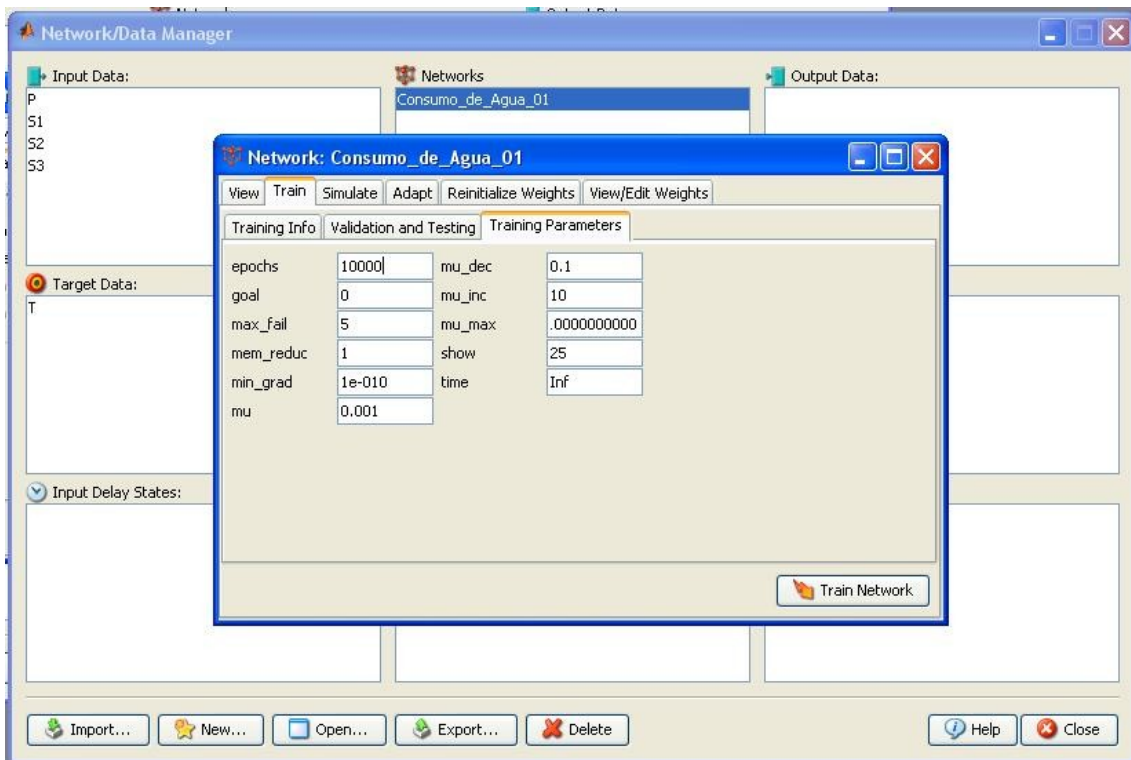


Figura 10 - Parâmetros de Treinamento

Após realizar todas as configurações de treinamento podemos pressionar o botão *Train Network* (Treinamento da Rede). Então surgirá um gráfico demonstrando. A figura 11 mostra o gráfico de desempenho da rede.

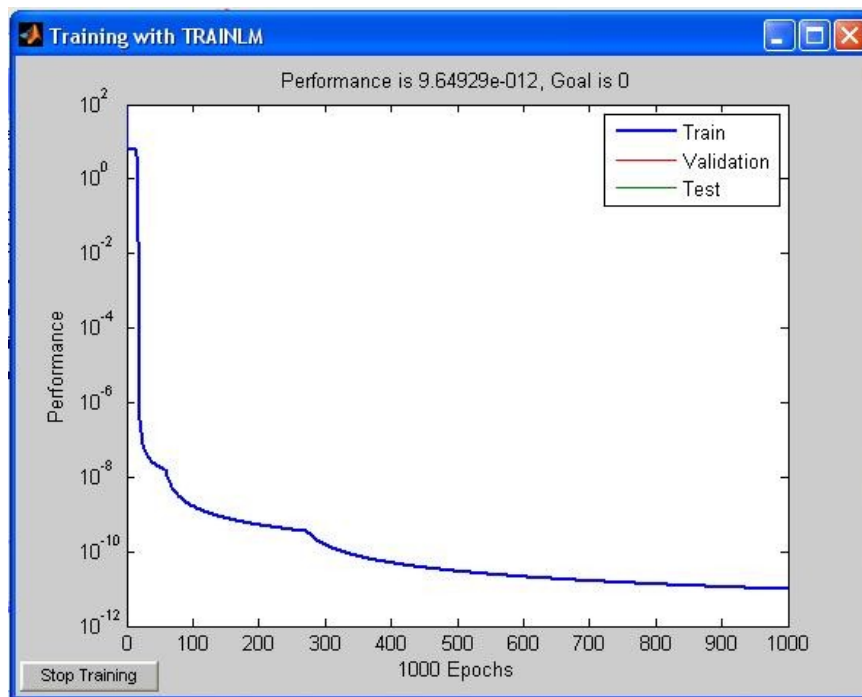


Figura 11 - Gráfico de Desempenho

4.2.4 Simulação

Com a rede treinada podemos clicar na aba simulação e fazer os testes com a entrada S1 inserida com valores de quatro meses. Também podemos dar um nome para a saída da simulação (*Simulation Results/Outputs*) para facilitar o entendimento ou diferenciar os resultados de saída quando trabalhamos com vários valores de entrada. Podemos observar na figura 12 a atribuição de as variável S1 para a simulação e o nome da saída “S1_output”.

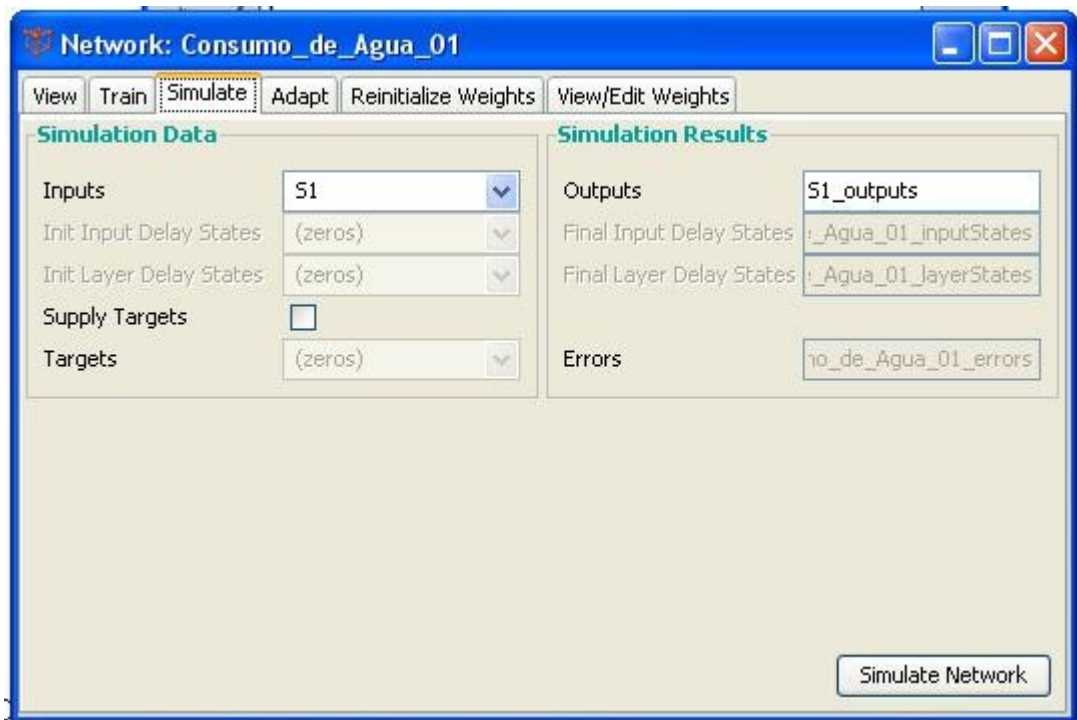


Figura 12 – Atribuição dos valores para a Simulação

Agora que os valores para a simulação já foram atribuídos podemos clicar no botão *Simulate Network*(Simular Rede) e voltar na tela inicial do NNTool para ver o resultado no quadro *Output Data*(Dados de Saída) e clicar na no nome da variável de saída especificada na etapa anterior. Basta Clicar duas vezes na variável para ver o resultado.

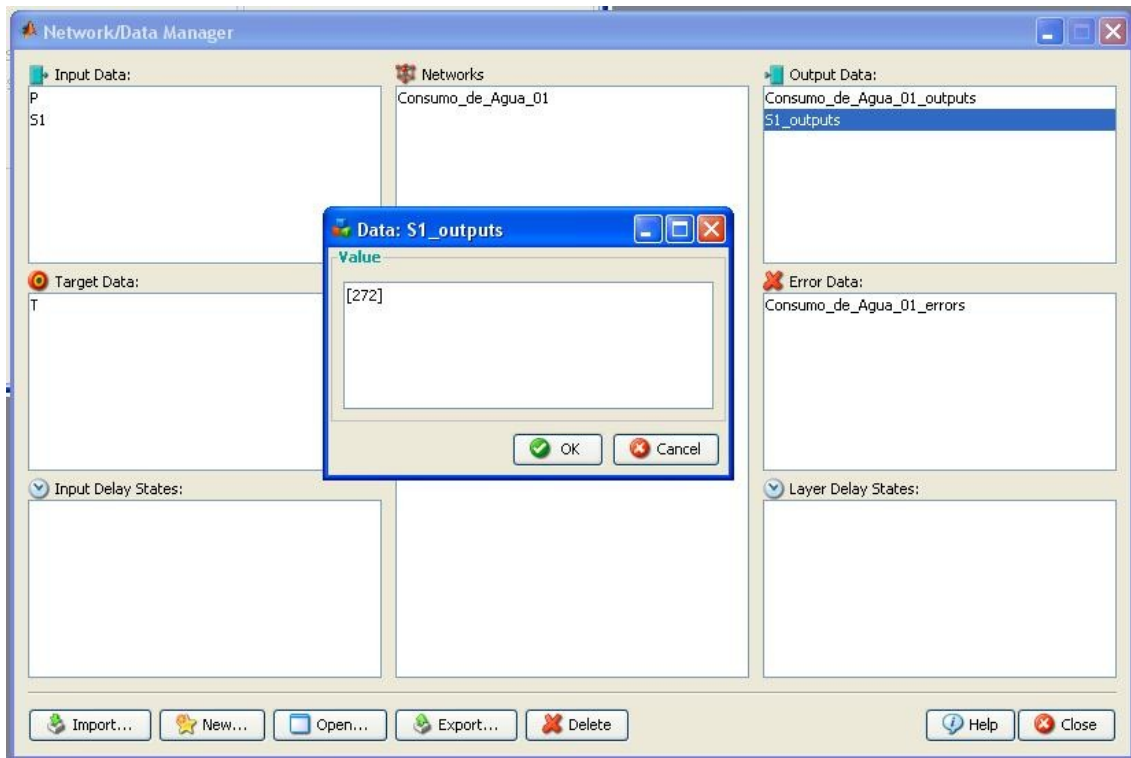


Figura 13 – Resultado da Simulação

5 CONCLUSÃO

Após a realização de todas as configurações, treinar a rede e realizar uma simulação, e finalmente o resultado. Como passamos os valores das medidas dos meses Outubro, Setembro, Agosto e Julho e o resultado encontrado foi o valor do mês de Novembro. Então nossa expectativa foi satisfeita já que o resultado encontrado foi o resultado esperado.

Referências bibliográficas

LUDWIG JR, Oswaldo; COSTA, Eduard M. **Redes Neurais: Fundamentos e Aplicações com Programas em C**. Rio de Janeiro: Editora Ciência Moderna Ltda. 2007;

www.ebah.com.br/apostila-matlab-pdf-a8868.html

Neural Networks Toolbox User's Guide:

http://www.cse.cuhk.edu.hk/corner/tech/doc/manual/matlab-5.3.1/help/pdf_doc/nnet/nnet.pdf