



DE SANTA CATARINA *CAMPUS JOINVILLE* DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SANTA
CATARINA *CAMPUS JOINVILLE*

CURSO: Engenharia Elétrica

MÓDULO:IX

PROFESSOR: Rodrigo Coral

UNIDADE CURRICULAR: Redes Neurais Artificiais

ALUNO: Elias Anzini Junior

REDES NEURASIS ARTIFICIAIS

RELATÓRIO TRABALHO 3

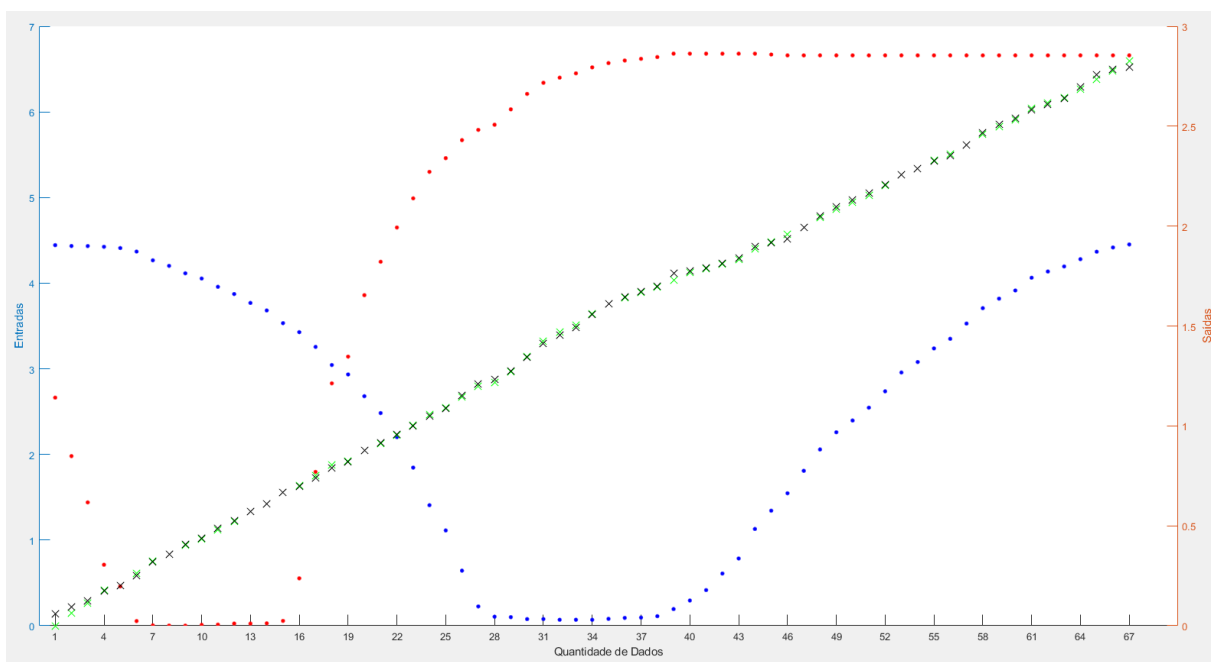
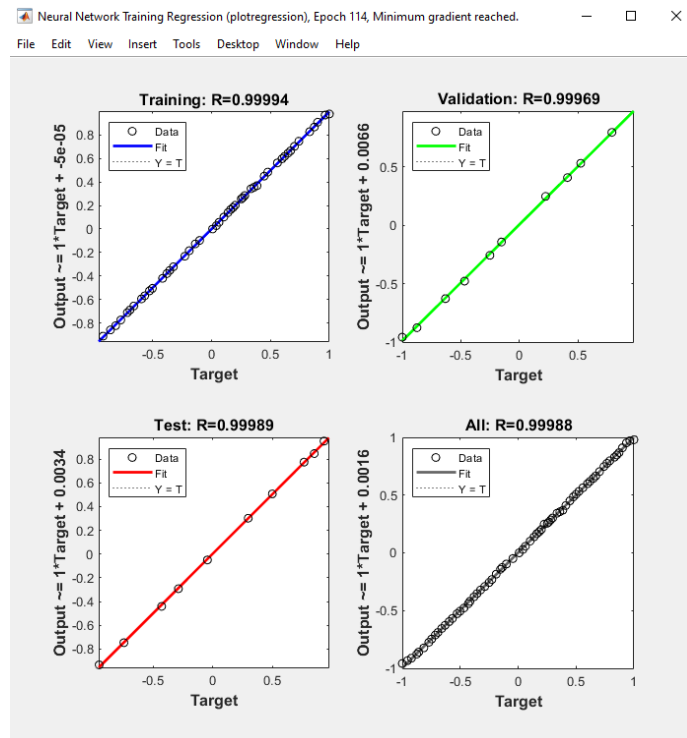
2.A) Treinamento de, no mínimo, 5 redes neurais do tipo feed-forward com diferentes topologias (quantidade de camadas, neurônios por camada, funções de ativação) utilizando todos os padrões da base de dados. Importante: explorar a flexibilidade que existe na definição da topologia.

Código empregado para treinamento e validação individual das redes:

```
1 %comandos de limpeza de tela e memória
2 - clear all;
3 - clc;
4
5 % Ler dados da planilha do Excel
6 - dadosV1 = xlsread('Trabalho 03 - MLP dados.xlsx', 'Sheet3', 'D4:BR4');
7 - dadosV2 = xlsread('Trabalho 03 - MLP dados.xlsx', 'Sheet3', 'D5:BR5');
8 - dadosD = xlsread('Trabalho 03 - MLP dados.xlsx', 'Sheet3', 'D6:BR6');
9
10 %Normalizando Dados
11
12 %Entrada
13 - [V1, norm_e] = mapminmax(dadosV1);
14 - [V2, norm_e] = mapminmax(dadosV2);
15 - E = [V1; V2]; % Concatenar as entradas
16 - [D, norm_s] = mapminmax(dadosD);
17
18 %Parâmetros de Treino
19 - RNA=newff(minmax(E), [3 4 1], {'tansig', 'tansig', 'tansig'}, 'trainlm');
20 - RNA.trainparam.epochs=10000;
21 - RNA.trainparam.goal=1e-10; %mse
22 - RNA.trainparam.min_grad=1e-4;
23 - RNA.trainparam.max_fail=10;
24
25 %aqui divide para fazer a validação
26 - RNA.divideFcn='dividerand';
27 - RNA=train(RNA,E,D);
28 - saida_RNA_norm = RNA(E);
29
30 %saida da rede desnormalizada
31 - saida_RNA=mapminmax('reverse',saida_RNA_norm, norm_s);
32 -
33
34 %Plotagem do gráfico
35 - figure(1)
36 - yyaxis left
37 - hold on
38 - plot(dadosV1, 'r.','MarkerSize', 12)
39 - plot(dadosV2, 'b.','MarkerSize', 12)
40
41 - ylabel('Entradas')
42 - yyaxis right
43 - ylabel('Saidas')
```

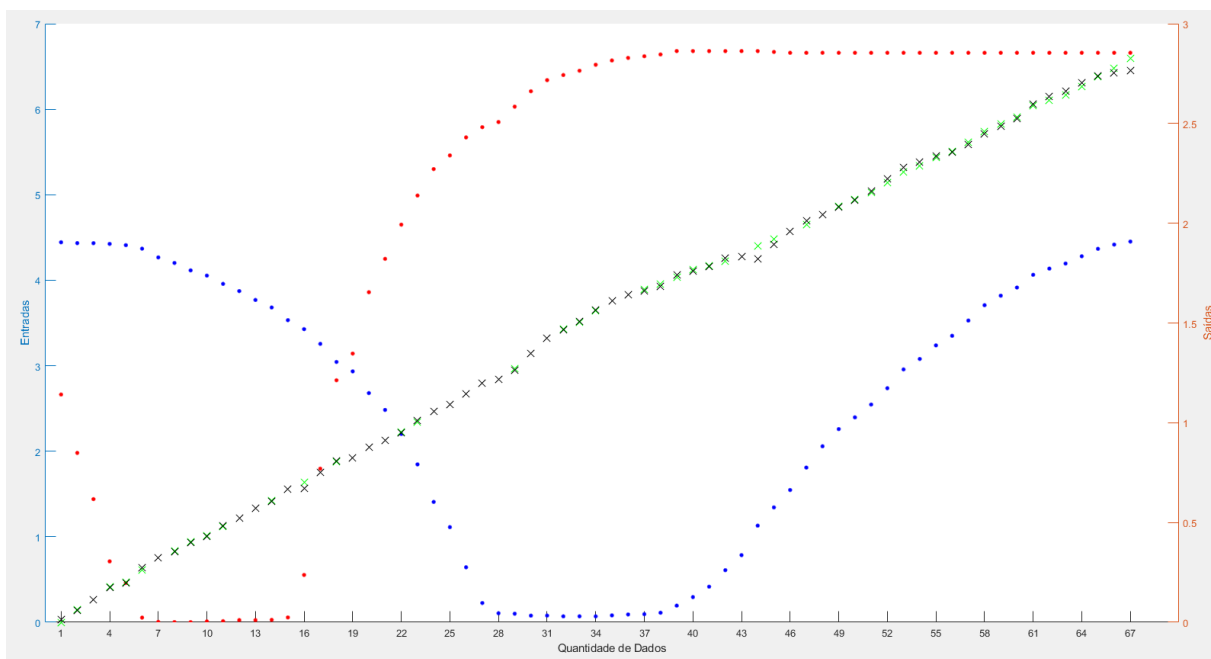
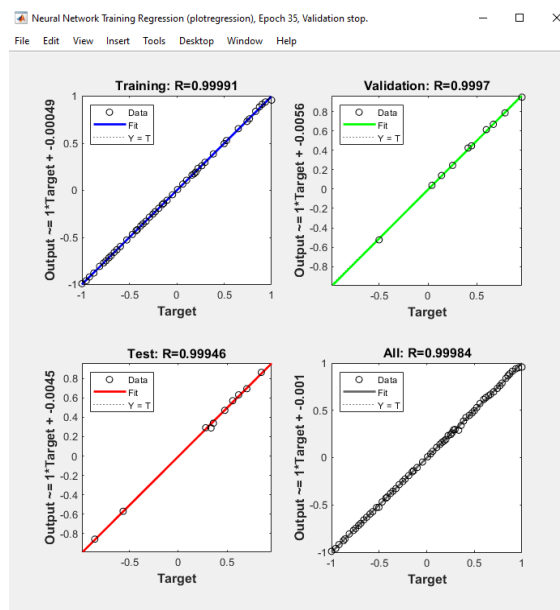
NETWORK 1: Para network 1, foram usados os valores padrões empregados em sala de aula para treino de rede.

```
%Parâmetros de Treino
RNA=newff(minmax(E), [3 4 1], {'tansig', 'tansig', 'tansig'}, 'trainlm');
RNA.trainparam.epochs=10000;
RNA.trainparam.goal=1e-10; %mse
RNA.trainparam.min_grad=1e-4;
RNA.trainparam.max_fail=10;
```



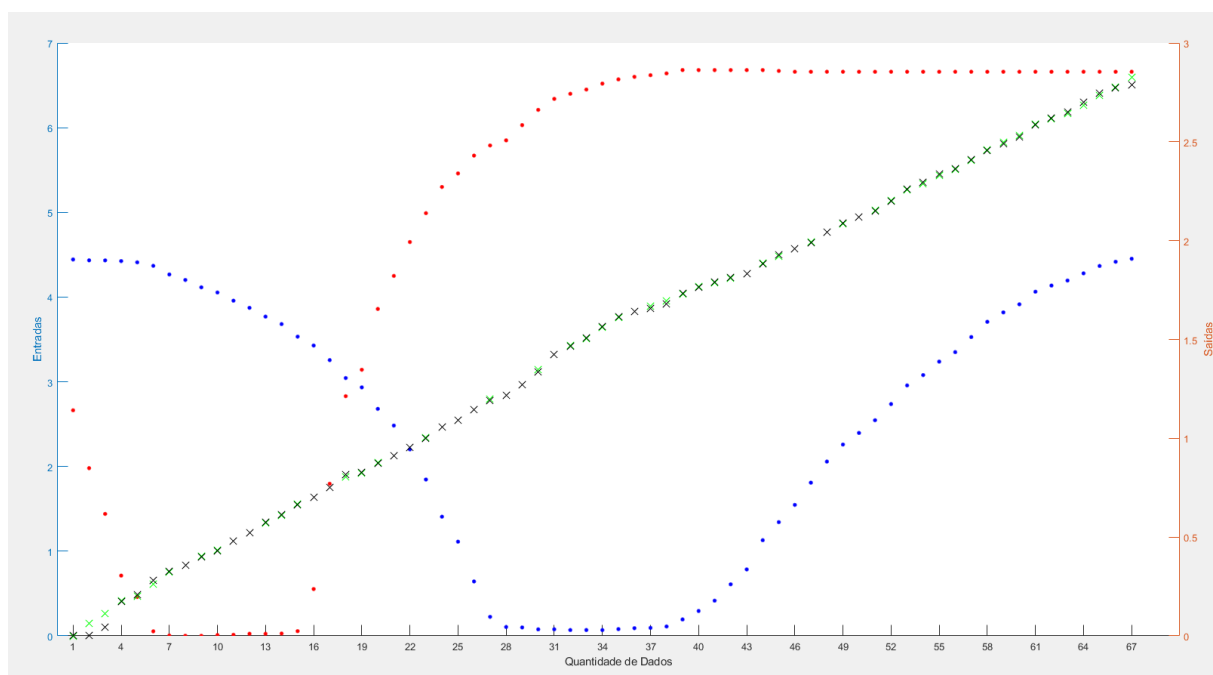
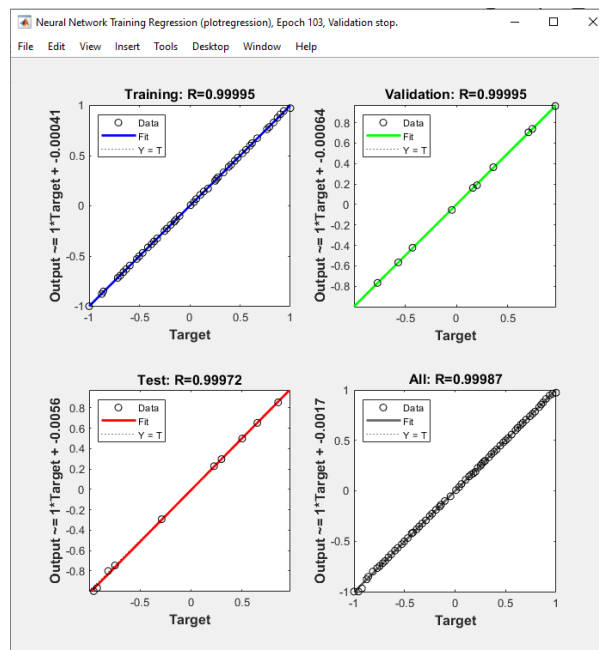
NETWORK 2: Para network 2, o número de neurônios nas camadas foi aumentado. Através desse teste, nota-se que para esses dados o aumento dos neurônios apenas aumenta o uso computacional e não implica em uma melhor performance da rede.

```
RNA=newff(minmax(E), [10 10 1], {'tansig', 'tansig', 'tansig'}, 'trainlm');
RNA.trainparam.epochs=10000;
RNA.trainparam.goal=1e-10; %mse
RNA.trainparam.min_grad=1e-4;
RNA.trainparam.max_fail=10;
```



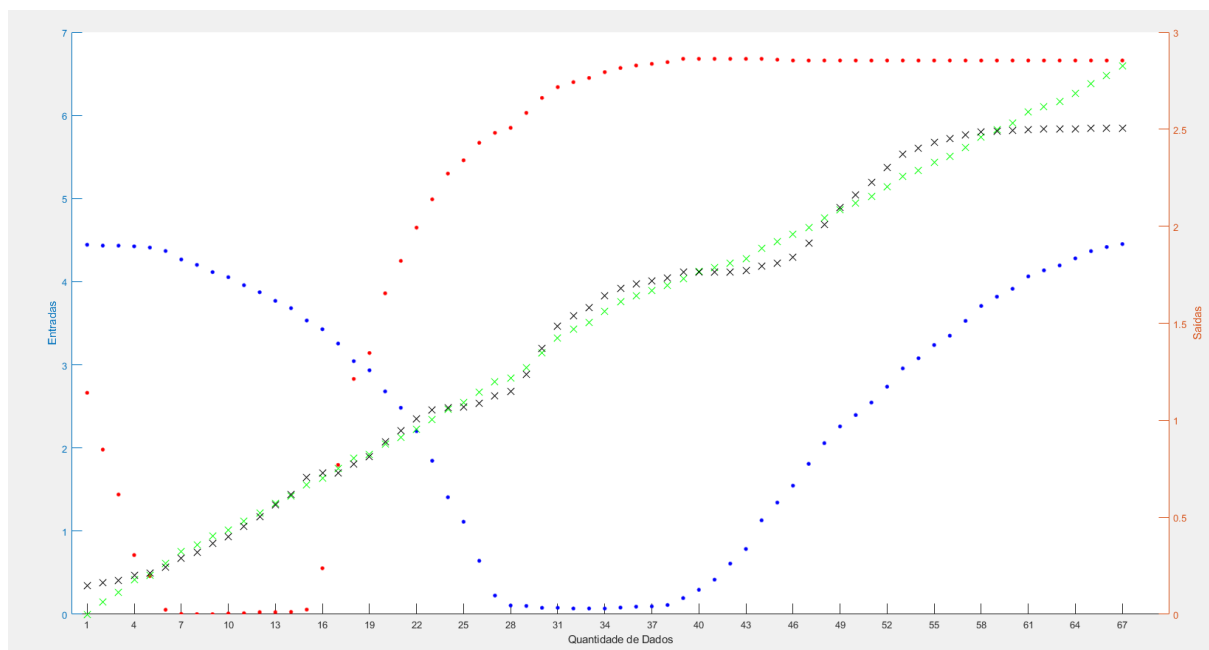
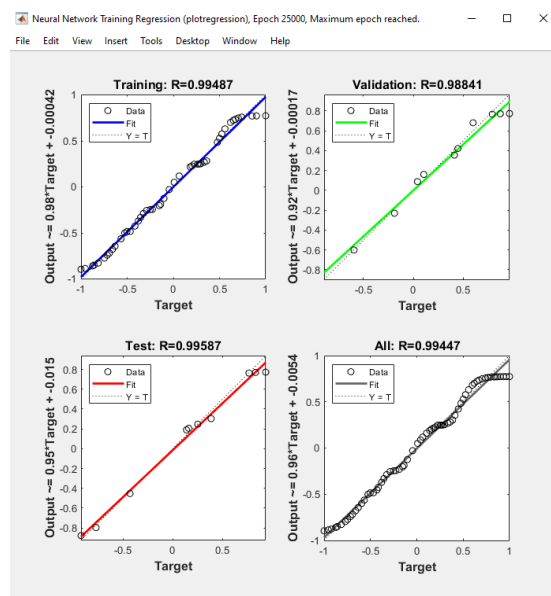
NETWORK 3: Para network 3, o número de camadas foi aumentado. Através desse teste, nota-se que para esses dados o aumento dos neurônios apenas aumenta o uso computacional e não implica em uma melhor performance da rede.

```
RNA=newff(minmax(E), [3 3 3 3 1], {'tansig', 'tansig', 'tansig', 'tansig', 'tansig'}, 'trainlm');
RNA.trainparam.epochs=10000;
RNA.trainparam.goal=1e-10; %mse
RNA.trainparam.min_grad=1e-4;
RNA.trainparam.max_fail=10;
```



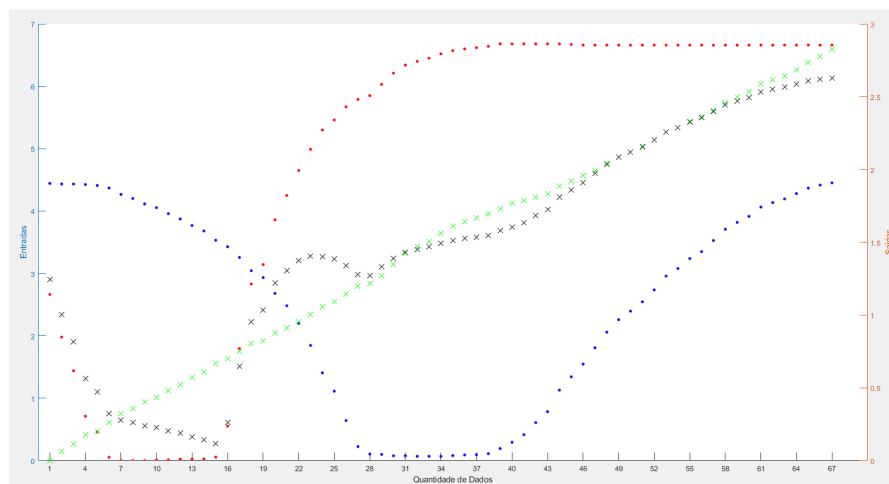
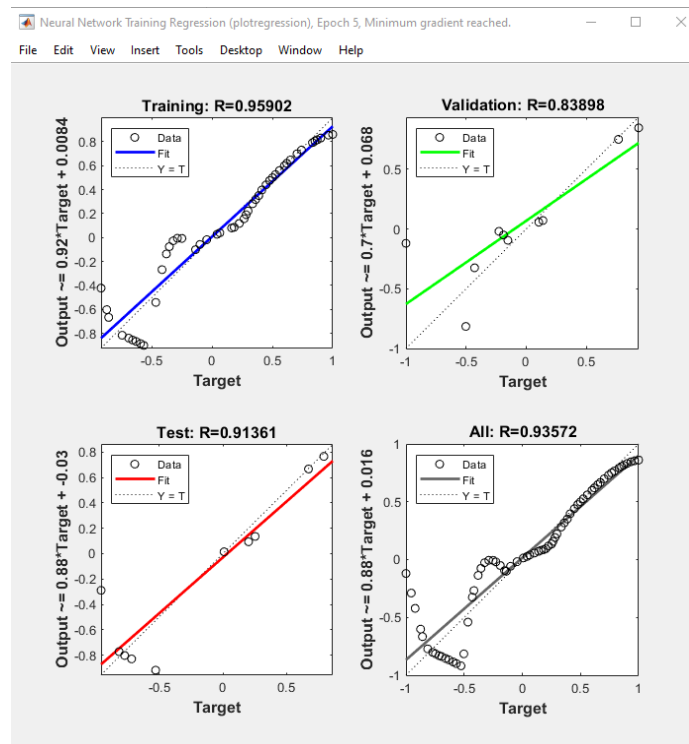
NETWORK 4: Para network 4, o método de treino aplicado foi o gradiente descendente. Para chegar a um resultado razoável, foi necessário aumentar o número de épocas e validation checks. Com isso, é possível notar que o método de treino Levenberg-Marquardt é muito melhor que o gradiente descendente.

```
RNA=newff(minmax(E), [3 4 1], {'tansig', 'tansig', 'tansig'}, 'traingd');
RNA.trainparam.epochs=25000;
RNA.trainparam.goal=1e-10; %mse
RNA.trainparam.min_grad=1e-4;
RNA.trainparam.max_fail=25;
```



NETWORK 5: Para network 5, foi utilizada a função de ativação purelin. Além disso, foi reduzido os parâmetros de validation check, mse goal e mínimo gradiente descendente.

```
%Parâmetros de Treino
RNA=newff(minmax(E), [3 4 1], {'purelin', 'purelin', 'purelin'}, 'trainlm');
RNA.trainparam.epochs=10000;
RNA.trainparam.goal=1e-3; %mse
RNA.trainparam.min_grad=1e-3;
RNA.trainparam.max_fail=2;
```

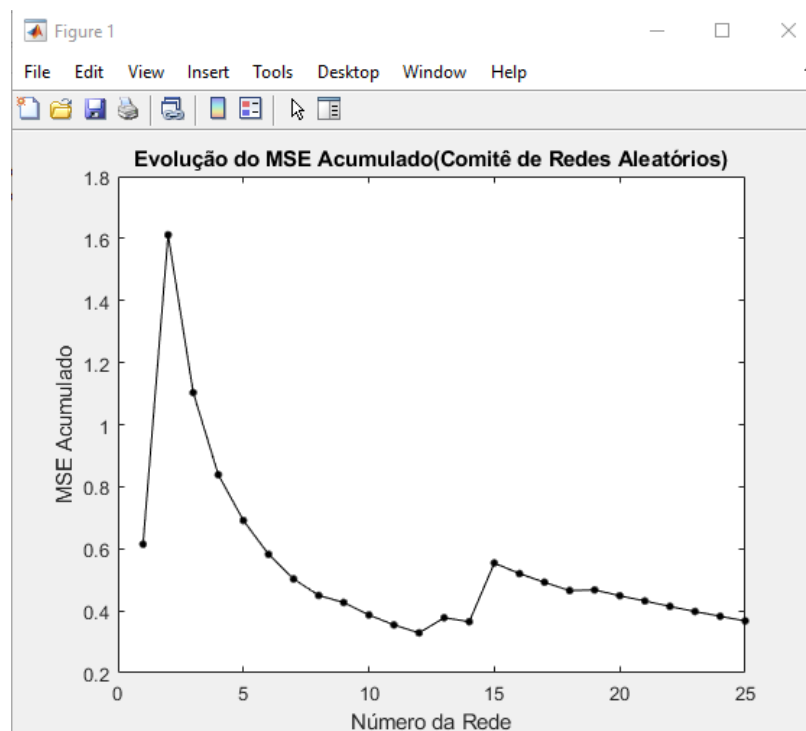


Topologia Empregada: Através dos testes realizados, nota-se um bom desempenho na network 1, 2 e 3. A topologia empregada para os próximos passos do trabalho é a 1, pois apresenta um desempenho similar às outras mas com menor uso computacional.

Para a topologia escolhida no item a deve ser realizado um estudo com foco na confiabilidade da rede neural.

i. Treinar, no mínimo, 5 redes neurais a partir de subconjuntos de dados, escolhidos aleatoriamente, da base de dados completa. A quantidade de dados escolhidos para os conjuntos de treinamento deve ser de 30% do total de padrões; o restante dos dados deve ser utilizado para teste das redes. Analisar o desempenho das redes neurais com foco na representatividade dos dados escolhidos aleatoriamente.

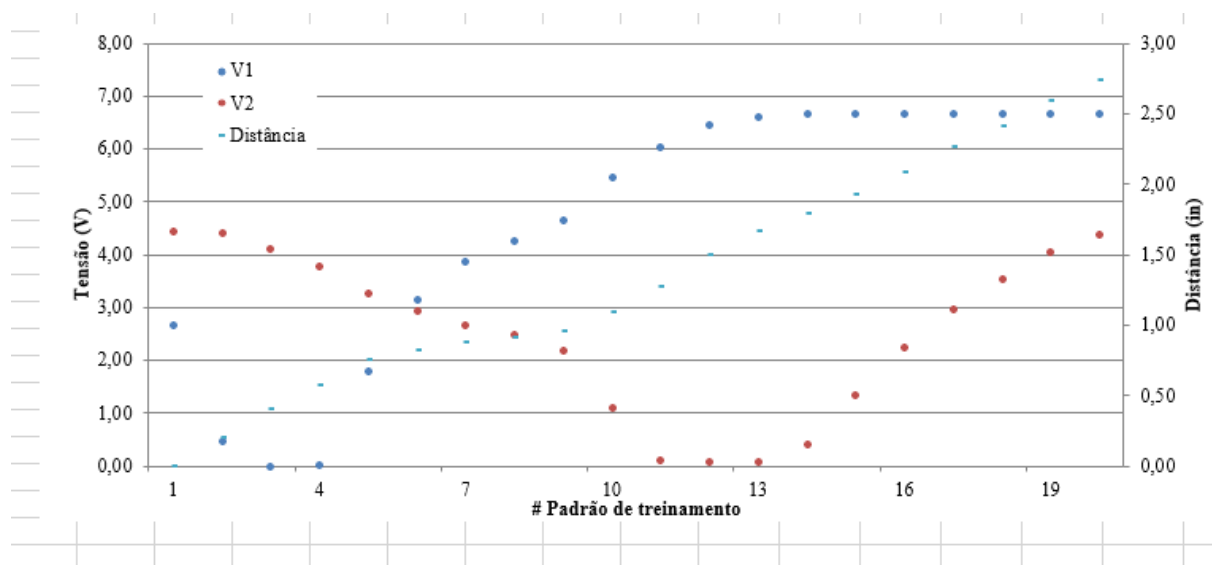
Os dados foram escolhidos de forma aleatória através da função `randperm`. Como pode-se notar, o gráfico do MSE estabilizou em aproximadamente 0,4. O número de redes utilizadas no comitê foi 25 redes.

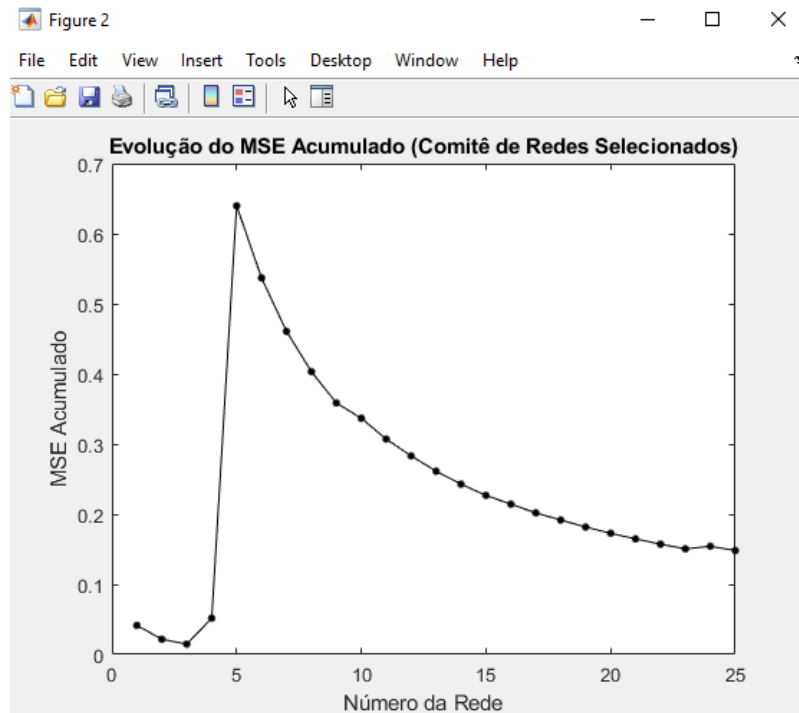


ii .Treinar 5 redes neurais com conjuntos de treinamento escolhidos manualmente de modo a evidenciar a importância da representatividade dos dados na descrição do processo. A quantidade de dados escolhidos para os conjuntos de treinamento deve ser de 30% do total de padrões. Analisar o desempenho das redes neurais com foco na representatividade dos dados escolhidos

Para escolha dos dados manuais, foram selecionados 20 dados para testes que compreendem toda a área do gráfico. Deste modo, evitando um possível erro que pode ocorrer de forma aleatória que seria selecionar dados apenas de uma região do gráfico. No ponto onde V1 e V2 se cruzam foram adicionados uma maior quantidade de dados para teste. A quantidade de redes no comitê escolhida foi de 25 redes.

V1	2,66	2,66	0,00	0,03	0,06	0,55	1,80	2,83	4,25	5,46	6,03	6,45	6,62	6,68	6,67	6,66	6,66	6,66	6,66
V2	4,44	4,44	4,12	3,77	3,53	3,43	3,26	3,04	2,48	1,11	0,10	0,07	0,09	0,42	1,34	2,26	2,96	3,53	4,06
D	0,00	0,00	0,40	0,57	0,67	0,70	0,75	0,81	0,91	1,09	1,27	1,50	1,67	1,79	1,92	2,09	2,26	2,41	2,59



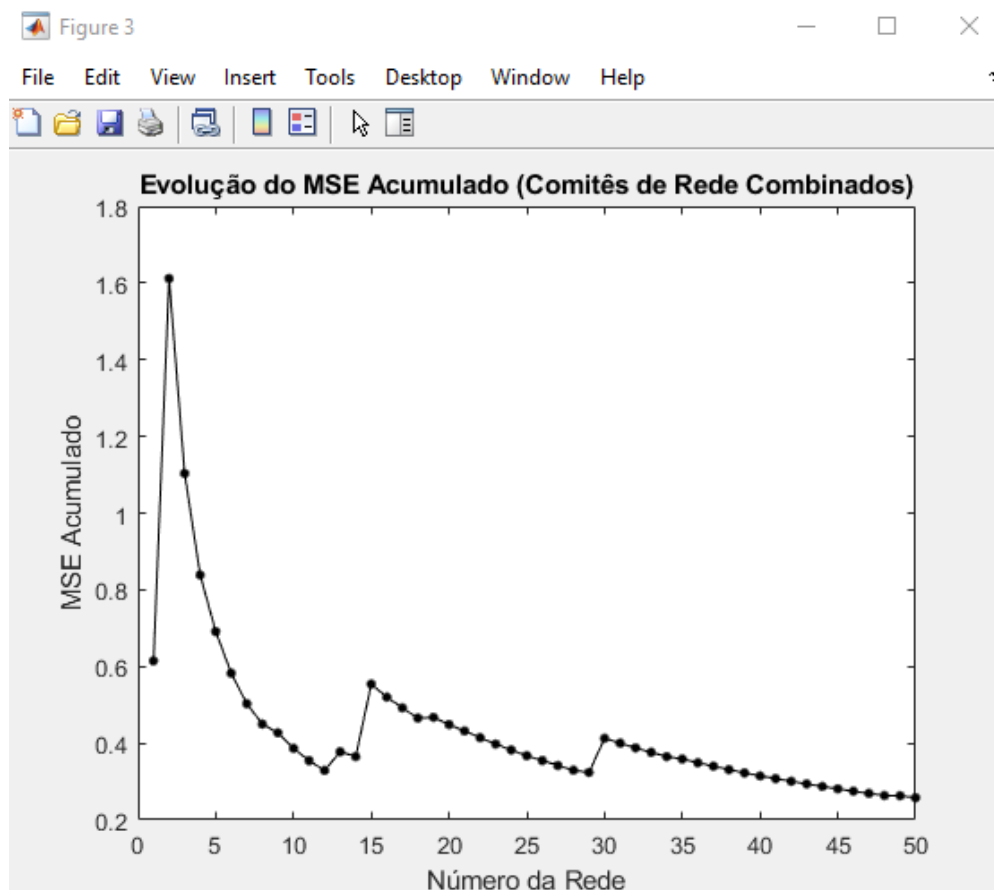


Como pode-se notar, o MSE começa a se estabilizar a partir da rede 20 e tem um resultado de aproximadamente 0,15.

Os resultados obtidos através da rede com dados escolhidos foram melhores do que com a rede aleatória.

iii .Criar um comitê com, no mínimo, 10 redes neurais (usar as 5 redes do item i desta alínea, e completar com mais 5 redes com os mesmos dados escolhidos aleatoriamente). Analisar o desempenho do comitê frente ao desempenho das redes neurais individuais.

Para essa atividade, foi empregado os comitês do exercício i e ii. Após isso, foi realizada a nova soma do MSE acumulado. O comitê de redes acumuladas não apresentou resposta melhor que a de dados selecionados. Isso pelo fato de possuir redes com dados aleatórios selecionados.



Código empregado para criação do comitê de redes com dados aleatórios:

```
*****-----ALEATÓRIOS-----*****

% Ler dados da planilha do Excel
dadosV1 = xlsread('Trabalho 03 - MLP dados.xlsx', 'Sheet3', 'D4:BR4');
dadosV2 = xlsread('Trabalho 03 - MLP dados.xlsx', 'Sheet3', 'D5:BR5');
dadosS = xlsread('Trabalho 03 - MLP dados.xlsx', 'Sheet3', 'D6:BR6');
dados = [dadosV1; dadosV2];

%Entrada
[E, norm_e] = mapminmax(dados);

%Saida
[S, norm_s] = mapminmax(dadosS);

%Pegando as amostras aleatoriamente
numAmostras = size(E, 2); % Obtém o número de colunas em E
indicesAleatorios = randperm(numAmostras);

numAmostrasTreino = round(0.3 * numAmostras);
numAmostrasTeste = numAmostras - numAmostrasTreino;

entradas_treino = E(:, indicesAleatorios(1:numAmostrasTreino));
saidas_treino = S(:, indicesAleatorios(1:numAmostrasTreino));

entradas_teste = E(:, indicesAleatorios(numAmostrasTreino+1:end));
saidas_teste = S(:, indicesAleatorios(numAmostrasTreino+1:end));

num_redes = 25;
saida_RNAs = cell(1,num_redes)
saida_RNAs_desnorm = cell(1,num_redes)

for i = 1:num_redes
    %Parâmetros de Treino
    RNA=newff(minmax(entradas_treino), [3 4 1], {'tansig', 'tansig', 'tansig'}, 'trainlm');
    RNA.trainparam.epochs=10000;
    RNA.trainparam.goal=1e-10; %mse
    RNA.trainparam.min_grad=1e-4;
    RNA.trainparam.max_fail=10;
    RNA.divideFcn = 'dividerand'

    %Execução da Rede
    RNA=train(RNA,entradas_treino,saidas_treino);
    saida_RNAs(i) = RNA(entradas_teste);
    saida_RNAs_desnorm(i)=mapminmax('reverse',saida_RNAs(i), norm_s);
end

saidas_teste_desnorm=mapminmax('reverse',saidas_teste, norm_s);
vetor_redes = 1:num_redes;

% Calculando o MSE acumulado
mseAcumulado = zeros(1, num_redes);

for k = 1:num_redes
    erros = saida_RNAs_desnorm{k} - saidas_teste_desnorm;
    mseAcumulado(k) = mean(erros.^2);
end

% Calculando o MSE médio acumulado
mseMediaAcumulada = zeros(1, num_redes);

for k = 1:num_redes
    mseMediaAcumulada(k) = mean(mseAcumulado(1:k));
end

% Plotando a evolução do MSE acumulado
figure;
plot(vetor_redes, mseMediaAcumulada, 'k.-', 'MarkerSize', 12);
xlabel('Número da Rede');
ylabel('MSE Acumulado');
title('Evolução do MSE Acumulado(Comitê de Redes Aleatórios)');
```

Código empregado para criação do comitê de redes com dados selecionados:

```
#####-----SELECIONADOS-----#####

% Ler dados da planilha do Excel
dadosV1_ii = xlsread('Trabalho 03 - MLP dados - Seleccionados.xlsx', 'Sheet3', 'D14:W14');
dadosV2_ii = xlsread('Trabalho 03 - MLP dados - Seleccionados.xlsx', 'Sheet3', 'D15:W15');
dadosS_ii = xlsread('Trabalho 03 - MLP dados - Seleccionados.xlsx', 'Sheet3', 'D16:W16');
dados_ii = [dadosV1_ii; dadosV2_ii];

dadosV1_teste_ii = xlsread('Trabalho 03 - MLP dados - Seleccionados.xlsx', 'Sheet3', 'D18:AX18');
dadosV2_teste_ii = xlsread('Trabalho 03 - MLP dados - Seleccionados.xlsx', 'Sheet3', 'D19:AX19');
dadosS_teste_ii = xlsread('Trabalho 03 - MLP dados - Seleccionados.xlsx', 'Sheet3', 'D20:AX20');
dados_teste_ii = [dadosV1_teste_ii; dadosV2_teste_ii];

%Entrada
[E_treino_ii, norm_e_ii] = mapminmax(dados_ii);
[E_teste_ii, norm_f_ii] = mapminmax(dados_teste_ii);

%Saida
[S_treino_ii, norm_s_ii] = mapminmax(dadosS_ii);

num_redes_ii = 25;
saida_RNAs_ii = cell(1,num_redes_ii)
saida_RNAs_desnorm_ii = cell(1,num_redes_ii)

for i = 1:num_redes_ii
    %Parâmetros de Treino
    RNA_ii=newff(minmax(E_treino_ii), [3 4 1], {'tansig', 'tansig', 'tansig'}, 'trainlm');
    RNA_ii.trainparam.epochs=10000;
    RNA_ii.trainparam.goal=1e-10; %mse
    RNA_ii.trainparam.min_grad=1e-4;
    RNA_ii.trainparam.max_fail=10;

    %Execução da Rede
    RNA_ii=train(RNA_ii,E_treino_ii,S_treino_ii);
    saida_RNAs_ii(i) = RNA_ii(E_teste_ii);
    saida_RNAs_desnorm_ii(i)=mapminmax('reverse',saida_RNAs_ii(i), norm_s_ii);
end

vetor_redes_ii = 1:num_redes_ii;

% Calculando o MSE acumulado
mseAcumulado_ii = zeros(1, num_redes_ii);

for k = 1:num_redes_ii
    erros_ii = saida_RNAs_desnorm_ii(k) - dadosS_teste_ii;
    mseAcumulado_ii(k) = mean(erros_ii.^2);
end

% Calculando o MSE médio acumulado
mseMediaAcumulada_ii = zeros(1, num_redes_ii);

for k = 1:num_redes_ii
    mseMediaAcumulada_ii(k) = mean(mseAcumulado_ii(1:k));
end

% Plotando a evolução do MSE acumulado
figure(2);
plot(vetor_redes_ii, mseMediaAcumulada_ii, 'k.-','MarkerSize', 12);
xlabel('Número da Rede');
ylabel('MSE Acumulado');
title('Evolução do MSE Acumulado (Comitê de Redes Seleccionados)');
```

Código para juntar o comitê de redes aleatórios e selecionados:

```
#####-CALCULANDO MSE ACUMULADO ENTRE OS 2 COMITES#####
vetor_redes_combinado = 1:(num_redes + num_redes_ii);

% Somando os MSEs acumulados dos dois comitês de rede
mseAcumulado_combinado = [mseAcumulado, mseAcumulado_ii];

% Calculando o MSE médio acumulado combinado
mseMediaAcumulada_combinada = zeros(1, num_redes + num_redes_ii);

for k = 1:(num_redes + num_redes_ii)
    mseMediaAcumulada_combinada(k) = mean(mseAcumulado_combinado(1:k));
end

% Plotando a evolução do MSE acumulado combinado
figure;
plot(vetor_redes_combinado, mseMediaAcumulada_combinada, 'k.-', 'MarkerSize', 12);
xlabel('Número da Rede');
ylabel('MSE Acumulado');
title('Evolução do MSE Acumulado (Comitês de Rede Combinados)');
```