



## Redes Neurais Artificiais

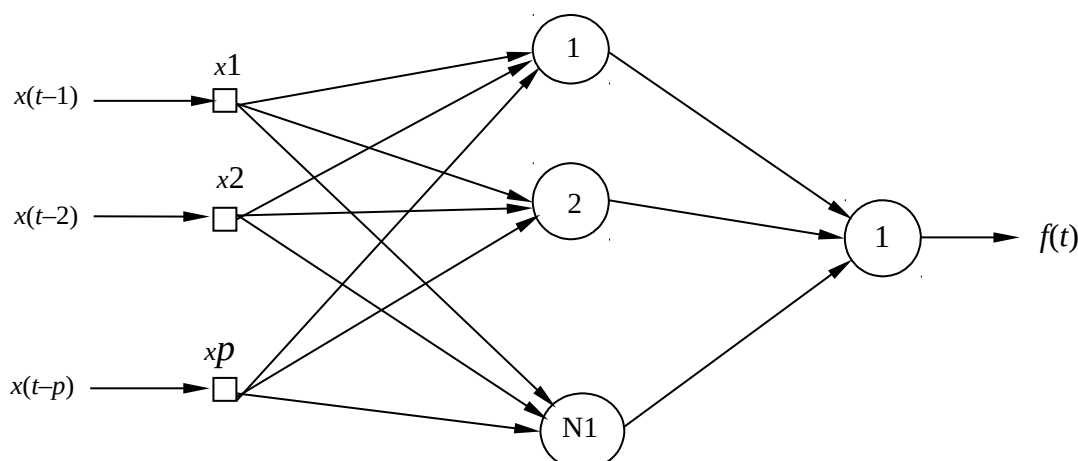
(Prof. Ivan Nunes da Silva)

**EPC-6**

**Nome: Luiz Gustavo Caobianco**

O preço de uma determinada mercadoria disposta para ser comercializada no mercado financeiro de ações possui um histórico de variação de valor conforme mostrado na tabela apresentada no Anexo.

Um pool de pesquisadores estará tentando aplicar redes neurais para tentar prever o comportamento futuro deste processo. Assim, pretende-se utilizar uma arquitetura perceptron multicamadas, com topologia “Time Delay” (TDNN), conforme mostrada na figura abaixo:



As topologias candidatas para serem adotadas no mapeamento do problema acima são especificadas como se segue:

**Rede 1** → 05 entradas ( $p = 05$ ) com  $N1 = 10$

**Rede 2** → 10 entradas ( $p = 10$ ) com  $N1 = 15$

**Rede 3** → 15 entradas ( $p = 15$ ) com  $N1 = 25$

Utilizando o algoritmo de aprendizagem *backpropagation com momentum* e os dados de treinamento apresentados no Anexo, realize as seguintes atividades:

1. Execute 3 treinamentos para cada rede perceptron acima, inicializando-se as matrizes de pesos em cada treinamento com valores aleatórios entre 0 e 1. Se for o caso, reinicie o gerador de números aleatórios em cada treinamento de tal forma que os elementos das matrizes de pesos iniciais não sejam os mesmos. Utilize a função de ativação logística para todos os neurônios, taxa de aprendizado  $\eta = 0.1$ , fator de momentum  $\alpha = 0.8$  e precisão  $\varepsilon = 0.5 \times 10^{-6}$ .



2. Registre os resultados finais desses 3 treinamentos, considerando-se cada uma dessas três topologias de rede, na tabela a seguir:

Treinamento	Rede 1		Rede 2		Rede 3	
	EQM	Épocas	EQM	Épocas	EQM	Épocas
1º (T1)	4.928672e-07	148	4.543946e-07	510	4.982219e-07	2084
2º (T2)	1.882650e-07	115	4.444851e-07	2	4.997940e-07	1804
3º (T3)	7.268661e-08	281	4.968539e-07	545	4.974780e-07	1856

3. Para todos os treinamentos efetuados no item 2, faça então a validação da rede em relação aos valores desejados apresentados na tabela abaixo. Forneça para cada treinamento o erro relativo médio entre os valores desejados e os valores fornecidos pela rede em relação a todas as amostras de teste. Obtenha também a respectiva variância.

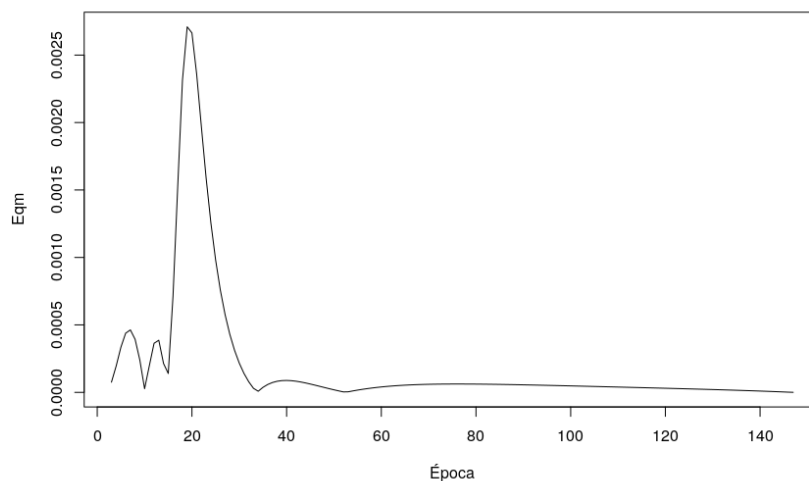
Amostra	$f(t)$	Rede 1			Rede 2			Rede 3		
		(T1)	(T2)	(T3)	(T1)	(T2)	(T3)	(T1)	(T2)	(T3)
$t = 101$	0.4173	0.1599	0.1534	0.0999	0.4299	0.3415	0.4283	0.6856	0.6687	0.6789
$t = 102$	0.0062	0.2471	0.2410	0.3838	0.2446	0.3242	0.2535	0.1845	0.1291	0.1208
$t = 103$	0.3387	0.1854	0.1848	0.1096	0.6774	0.3441	0.6778	0.2967	0.3243	0.3110
$t = 104$	0.1886	0.7125	0.7376	0.7346	0.3378	0.3297	0.3396	0.2970	0.3663	0.3922
$t = 105$	0.7418	0.1494	0.1546	0.0900	0.2258	0.3408	0.2225	0.1227	0.0810	0.0739
$t = 106$	0.3138	0.2696	0.2600	0.2518	0.1243	0.3275	0.1208	0.7094	0.7158	0.7237
$t = 107$	0.4466	0.2471	0.2411	0.1338	0.3659	0.3378	0.3800	0.1736	0.1602	0.1613
$t = 108$	0.0835	0.1462	0.1474	0.3522	0.0949	0.3285	0.0920	0.3339	0.3198	0.3095
$t = 109$	0.1930	0.2085	0.2068	0.1176	0.8032	0.3407	0.8170	0.4232	0.4516	0.4684
$t = 110$	0.3807	0.6749	0.7032	0.7122	0.0648	0.3314	0.0734	0.4101	0.3427	0.3368
$t = 111$	0.5438	0.2076	0.2185	0.1364	0.0888	0.3339	0.0843	0.1087	0.1610	0.1558
$t = 112$	0.5897	0.3207	0.3013	0.2277	0.4638	0.3345	0.4657	0.1327	0.1331	0.1294
$t = 113$	0.3536	0.3310	0.3204	0.1529	0.1242	0.3339	0.1383	0.1812	0.1518	0.1625
$t = 114$	0.2210	0.1096	0.1169	0.3242	0.6545	0.3344	0.6527	0.8992	0.8566	0.8715
$t = 115$	0.0631	0.2042	0.2012	0.1204	0.4320	0.3340	0.4447	0.0538	0.0412	0.0361
$t = 116$	0.4499	0.6391	0.6686	0.7043	0.1440	0.3344	0.1394	0.0648	0.0991	0.1015
$t = 117$	0.2564	0.2130	0.2318	0.1748	0.2066	0.3341	0.1930	0.7354	0.6891	0.7260
$t = 118$	0.7642	0.3769	0.3414	0.2032	0.3103	0.3343	0.3420	0.3013	0.1414	0.1309
$t = 119$	0.1411	0.4282	0.4058	0.1648	0.0791	0.3341	0.0751	0.2777	0.4008	0.3704
$t = 120$	0.3626	0.1027	0.1119	0.2950	0.8161	0.3343	0.8362	0.0878	0.1393	0.1476
Erro Relativo Médio:		2.6700	2.6200	3.8300	2.9200	3.2500	3.0100	2.3700	1.9700	1.9200
Variância:		0.0700	0.0700	0.1020	0.1040	0.0400	0.1050	0.1200	0.0120	0.1300

4. Para cada uma das topologias apresentadas na tabela acima, considerando-se ainda o melhor treinamento {T1, T2 ou T3} realizado em cada uma delas, trace então o gráfico dos valores de erro quadrático médio (EQM) em função de cada época de treinamento. Imprima os três gráficos numa mesma folha de modo não superpostos.



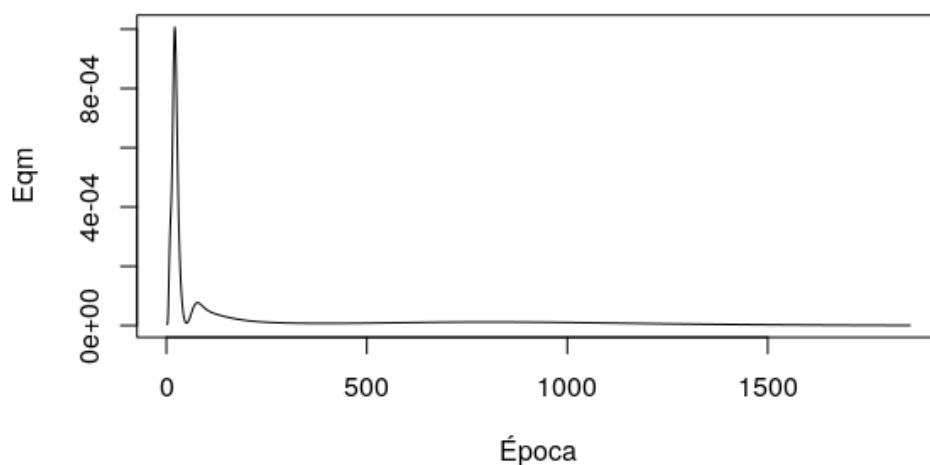
### Topologia 1

Erro Quadrático Médio



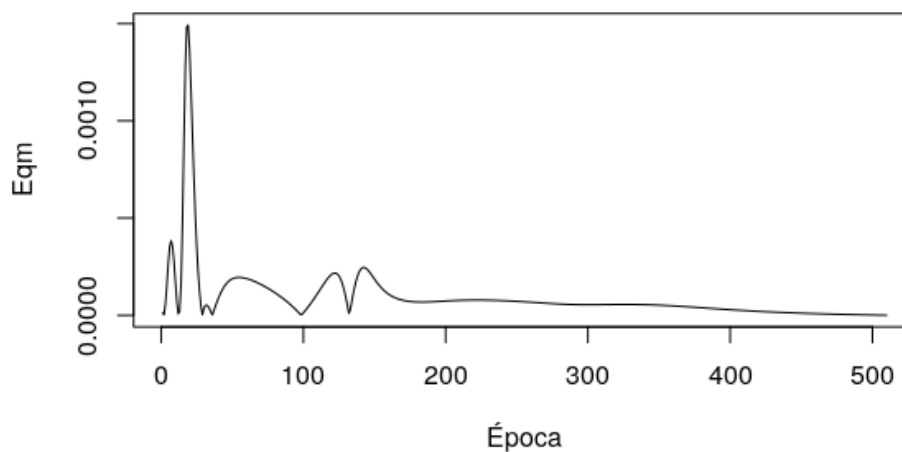
### Topologia 2

Erro Quadrático Médio



### Topologia 3

Erro Quadrático Médio



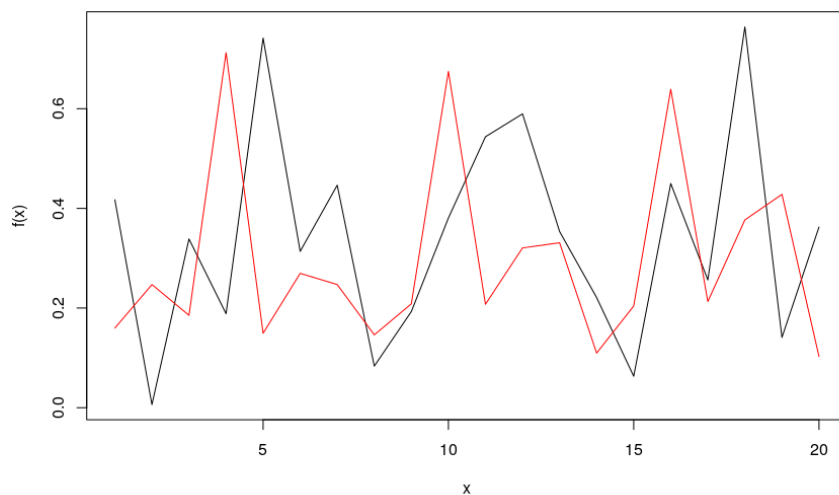


5. Para cada uma das topologias apresentadas na tabela acima, considerando-se ainda o melhor treinamento {T1, T2 ou T3} realizado em cada uma delas, trace então o gráfico dos valores desejados e dos valores estimados pela respectiva rede em função do domínio de operação assumido ( $t=101..120$ ). Imprima os três gráficos numa mesma folha de modo não superpostos.



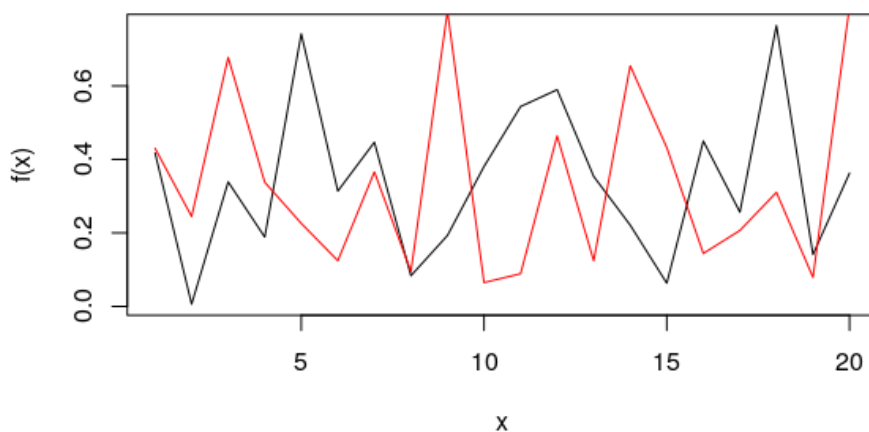
### Topologia 1

Comparação entre função exata e aproximada



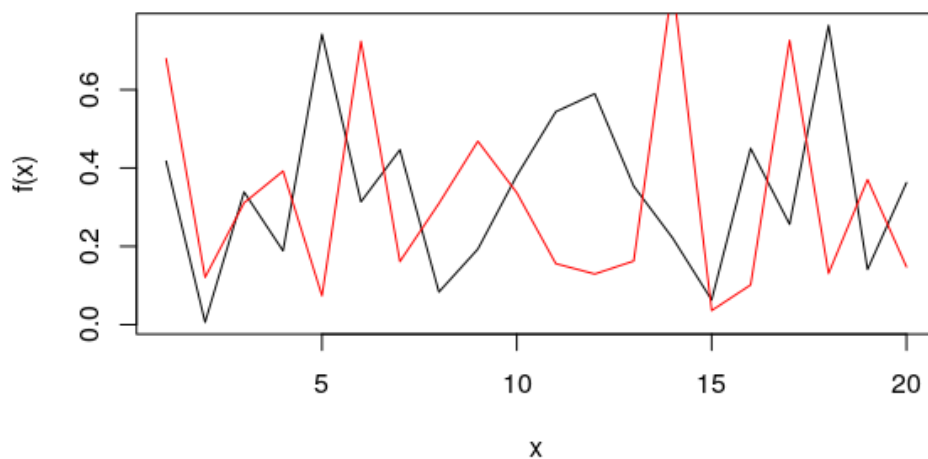
### Topologia 2

Comparação entre função exata e aproximada



### Topologia 3

Comparação entre função exata e aproximada





6. Baseado nas análises dos itens acima, indique então qual das topologias candidatas {Rede 1, Rede 2 ou Rede 3}, e que com qual configuração final de treinamento {T1, T2 ou T3}, seria a mais adequada para realização de previsões neste processo.

- **Resposta – Levando em consideração os erros relativos médios apresentados pelas topologias em discussão, a melhor opção seria a terceira topologia, que oferece uma aproximação melhor da função original.**

7. Em relação aos métodos de treinamento que são variantes do algoritmo *backpropagation*, investigue e comente sobre as principais características e vantagens dos seguintes algoritmos:

- a. Algoritmo de treinamento Resilient-Propagation (Rprop).
- b. Algoritmo de treinamento Levenberg-Marquardt (LM).

- **Resposta – O método Rprop leva em consideração apenas o sinal do gradiente, em detrimento ao método *backpropagation*, que leva em consideração o valor do gradiente. A vantagem disso é que, caso o valor do gradiente seja pequeno, a convergência será mais lenta. Já com o algoritmo Rprop, a taxa de aprendizagem pode ser mudada dinamicamente. Por exemplo, se os sinais do gradiente de duas iterações diferentes forem iguais, pode-se aumentar a taxa de aprendizagem, pois o algoritmo ainda está descendendo até o mínimo. Caso o sinal seja diferente, deve-se então diminuir a taxa de aprendizagem, de forma que o algoritmo continua descendo, mas de forma mais suave.**

**Já o algoritmo de Levenberg-Marquardt leva em consideração a matriz Jacobiana, e fornece uma convergência de 10 a 100x mais rápida que o tradicional *backpropagation*. Entretanto, deve-se atentar para o fato de que uma matriz  $J(z)$  mal condicionada pode acarretar em problemas de convergência.**