

Avaliação Prática de Redes Neurais para o Acasalamento de Gado Nelore¹

Rômulo B. Franco, Weber Martins
Lab. de Sistemas Inteligentes
Escola de Engenharia Elétrica - Pça Universitária s/n
E-mails: romulo, weber@eee.ufg.br

Abstract

This paper presents a system designed to predict the best bull should be chosen in order to obtain the best result given information on a specific cow. Learning is conducted by using data from real work done by specialists on multilayer perceptron networks and variants of backpropagation algorithm. The main goal was to find the best single neural network the could perform the task with confidence. This study provided some interesting results on the efficiency of backpropagation standard, with momentum and employing the technique of weight decay. Many factors were varied, including number of cycles, learning rate, momentum rate, topology, etc.

1. Introdução

Em primeiro lugar, cabe-nos mostrar a importância da tarefa sob análise. Geralmente, o acasalamento de gado é conduzido por especialista que determina o sêmen a ser utilizado em cada vaca num processo de inseminação artificial. O conhecimento de qual acasalamento deve ser empregado para cada vaca em particular decorre em grande parte da experiência do profissional. Frequentemente, presencia-se, mesmo em exposições agropecuárias, empregando juízes presumidamente bastante especializados, opiniões bastante divergentes sobre a qualidade de cada reprodutor. O cenário é ainda mais complicado pelo interesse econômico na fama de reprodutores para o sucesso da venda de seu sêmen.

O desenvolvimento da qualidade do gado nelore para o Brasil é incontestável. Trata-se do tipo de gado mais difundido e, comparando com a produtividade do gado holandês, por exemplo, muito resta por fazer.

Para sermos mais precisos, encontra-se em execução o Programa Nacional de Melhoramento de Gado Nelore (coordenado pela USP de Ribeirão Preto). Este programa, entretanto, utiliza-se apenas de técnicas estatísticas para tentar modelar processos ainda básicos dentro do objetivo final (como o acompanhamento da curva de crescimento de um animal, por exemplo).

Este artigo mostra estudos empíricos realizados com um grande número de simulações no sentido de melhor sintonizar a arquitetura multilayer perceptron (MLP) e o tipo de backpropagation mais adequado.

Outros trabalhos foram realizados com proposta semelhante, porém empregando outros modelos, como LVQ (Learning Vector Quantization) e SOM (Self-Organizing Maps) [1] e Lógica Nebulosa [2]. Apenas em [3], usa-se MLP, com uma metodologia diferente da aqui empregada. Há, também, um trabalho na área de gado holandês sobre aperfeiçoamento de gado no Canadá [4].

2. Metodologia

2.1. Os dados

A massa de dados fornecida foi um conjunto de 297 amostras, onde para cada vaca foram anotadas 10 características importantes para descrevê-las e qual, de um conjunto de 5 touros, o que melhor apresentou resultados. Um exemplo pode ser visto na Tabela 1 (ver a seguir).

Tabela 1: Estrutura dos registros de acasalamento

vaca		touro	
aspecto	valor	identificação	valor
C1	2	1	0
C2	1	2	0
C3	3	3	0
C4	4	4	1
C5	2	5	0
C6	4		
C7	4		
C8	4		
C9	2		
C10	3		

As características C1 a C10 são pesos atribuídos aos fatores descritos abaixo:

- C1 – Pigmentação
- C2 – Musculosidade

¹ Este trabalho conta com apoio da Fundação de Amparo à Pesquisa (FUNAPE).

- C3 – Estrutura
- C4 – Raça
- C5 – Conformação
- C6 – Onfalo
- C7 – Ganho de Peso
- C8 – Tempo de Prenhez
- C9 – Desmame
- C10 – Média ponderada dos anteriores

Sob ponto de vista da Veterinária, o presente estudo tenta determinar a real importância desta modelagem de vaca para o sucesso do acasalamento. Tal modelagem não é trivial, sendo um dos principais objetivos do Programa Nacional de Melhoramento de Gado Nelore.

No Touro é colocado 1 para o que melhor desempenho com as características das vacas (vencedor) e 0 para os demais.

2.2. As Redes Neurais

As redes escolhidas foram multilayer perceptrons (feedforward) com algoritmos de treinamento backpropagation padrão (standard), momentum e weight decay [5].

A arquitetura das mesmas foram estabelecidas obedecendo as classes de entrada e saída. Desta maneira, variamos apenas as camadas escondidas. Os modelos treinados estão descritos na Tabela 2 (ver abaixo).

Tabela 2: Arquiteturas Testadas

Modelo	Entrada	Escondida	Saída
01	10	10	5
02	10	10x10	5
03	10	5	5
04	10	5x5	5

2.3. O Treinamento

Para o treinamento foi utilizado o Simulador Neural SNNS (Stuttgart Neural Network Simulator).

As 297 classes foram divididas, proporcionalmente para cada touro vencedor, em dois conjuntos, um de treinamento com 198 amostras e outro de teste com 99.

O conjunto foi feito em numerosas simulações, resultado da combinação entre os diversos parâmetros abaixo:

- Tipo da situação, correspondendo a cada uma das quatro arquiteturas acima descritas.
- Quantidade de ciclos de treinamento, variando de 50 a 6400 ($50 * 2^n$, onde $n = 0, 1, 2 \dots 7$).
- Fator de aprendizado, η , sendo utilizados os valores 0.1, 0.2, 0.4 e 0.8.

- Fator de momentum, para o algoritmo de treinamento Backpropagation com Momentum, μ , assumindo os valores de 0.1, 0.2, 0.4 e 0.8.
- Fator de decaimento, para o algoritmo de treinamento Backpropagation com Weigth Decay, μ , sendo utilizados os valores 0.005, 0.02, 0.08 e 0.5.

No total, foram feitas 1152 simulações com os parâmetros acima, e um conjunto de inúmeras outras simulações guias para determinar os caminhos a serem adotados.

3. Resultados

Para apresentação de resultados, foi definido o grau de acerto da rede em 4 categorias, a seguir :

- R1: é considerado vencedor o de maior valor de saída (vencedor), sem levar em conta os demais.
- R2: é considerado vencedor o que tem um valor, pelo menos, 10% superior ao segundo colocado.
- R3: da mesma maneira de R2, porém com valor pelo menos 20% superior ao segundo.
- R4: o vencedor tem que ter valor pelo menos o dobro do segundo (50%).

O modelo de backpropagation standard (ver Figura 1) apresentou desempenhos similares para taxas de aprendizagem pequenas. Os resultados para $\eta=0.2$ foram excluídos da figura por apresentarem valores totalmente dentro da expectativa. Para $\eta=0.8$, o sistema não conseguiu convergir dado às fortes variações em cada ciclo. Neste modelo foram realizadas 32 simulações por arquitetura. Na Figura 1, os valores refletem o valor médio de cada uma, cabendo ressaltar que tais valores foram semelhantes para as mesmas configurações

O modelo de backpropagation momentum foi o que apresentou melhor performance, principalmente para valores de η e μ pequenos. Ao todo foram realizadas 128 simulações por arquitetura. Cabe dizer, sinteticamente, que as arquiteturas com mais neurônios, como já se podia esperar, demandou mais ciclos para atingir um bom desempenho, dado ao maior número de graus de liberdade (parâmetros de ajuste) destes sistemas.

Como pode ser observado na Figura 2, há um gráfico comparativo entre as quatro arquiteturas, mostrando em cada uma a classe de resultados. Nesta apresentação foi utilizados os valores de η e μ em que as redes obtiveram melhor desempenho, ou seja, 0,2 e 0,1 respectivamente. Resultados mais detalhados podem ser encontrados em [5].

O modelo de backpropagation weigth decay em todas as simulações, 128 por arquitetura, não

apresentou em nenhuma delas um quadro de estabilidade superior a 40% de acertos. Também foram testadas outras arquiteturas e disposições das classes de treinamento, sem obtenção de êxito.

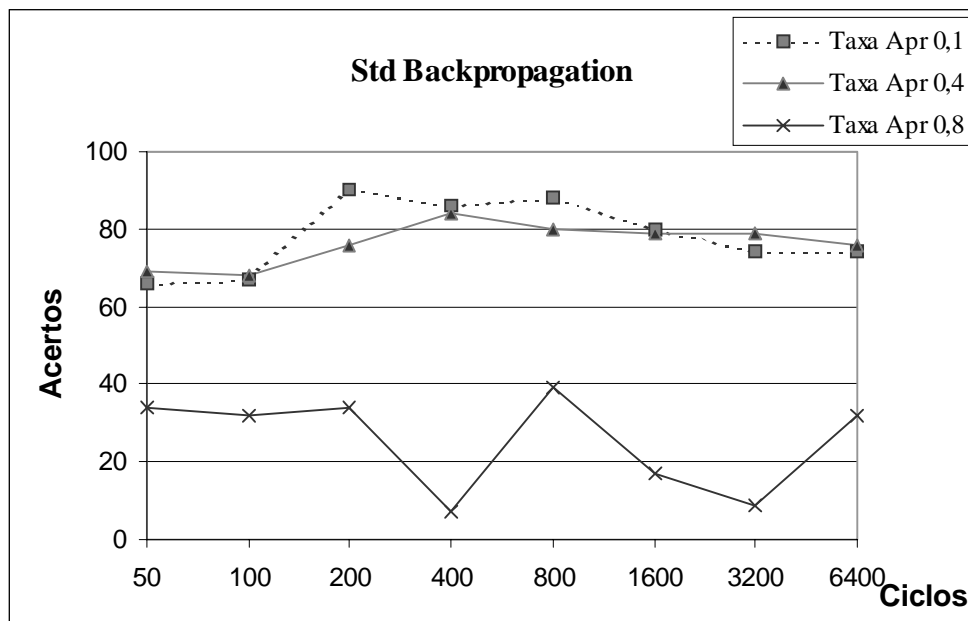


Figura 1:Desempenho do algoritmo de back-propagation.

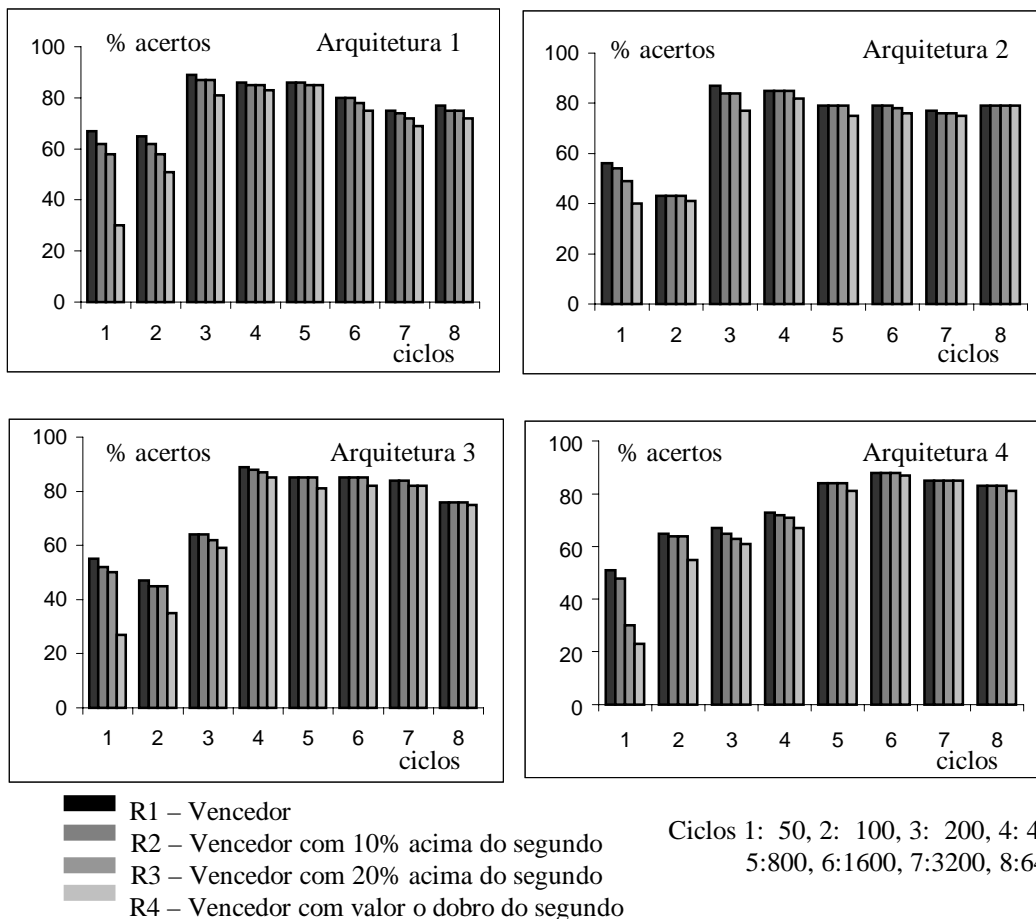


Figura 2: Percentagem de Acertos por Ciclos de Treinamento nas Diversas Arquiteturas (ver Tabela 2).

A Tabela 3, abaixo, apresenta um estudo sintético comparativo entre os diversos algoritmos de treinamento. Os resultados foram agrupados de acordo com as melhores performances obtidas. Estão apresentadas variações de cada arquitetura de acordo com os parâmetros fornecidos, dando uma idéia do comportamento geral de todo o processo. Os valores de máximo e mínimo apresentados são, respectivamente, a média dos mínimos e a média dos máximos obtida para para R1 (vencedor) e R4 (vencedor dobro do segundo). Desta maneira, a *média* considerando o vencedor pelo esquema R4, para a arquitetura A1, com 100 ciclos, e backpropagation standard foi de 51/99.

Conforme a citada tabela, nota-se que, no geral, o algoritmo backpropagation com momentum teve um bom desempenho. Ele conseguiu definir bem o vencedor, com um número relativamente pequeno de ciclos, além de uma boa estabilidade (variabilidade, dispersão). O algoritmo standard backpropagation também apresentou uma boa convergência com algumas poucas simulações tendo um desempenho um pouco melhor que o backpropagation com momentum. Já o algoritmo backpropagation com weight decay não apresentou nenhum tipo de convergência ou garantia de integridade dos resultados.

Tabela 3: Quadro Comparativo entre os Algoritmos

Modelo	Arquitetura	100 ciclos		200 Ciclos		400 Ciclos		800 Ciclos	
		R4	R1	R4	R1	R4	R1	R4	R1
Standard Backpropagation	A1	51	67	82	90	82	86	87	88
	A2	39	48	66	79	73	81	78	86
	A3	7	69	19	68	37	76	17	84
	A4	32	68	44	80	52	82	55	84
Backpropagation com Momentum	A1	51	65	81	89	83	86	85	86
	A2	41	43	77	87	82	85	75	79
	A3	35	47	59	64	85	89	81	85
	A4	55	65	61	67	67	73	81	84
Backpropagation com Weight Decay	A1	0	29	0	29	0	29	0	38
	A2	0	38	0	10	0	38	0	13
	A3	0	10	15	38	0	10	0	28
	A4	1	31	0	29	0	28	3	33

4. Conclusões

Para finalizar, este trabalho foi motivado pelo baixo desempenho alcançado nas tentativas anteriores de aplicação de redes neurais ao problema em questão, onde o desempenho máximo situou-se na faixa de 75% [1, 2, 3]. O sistema proposto atingiu índices de 85% de acertos, mostrando-se, portanto, melhor.

Apesar do sentido prático do trabalho, procurou-se aproveitar a situação para testar hipóteses sobre as condições mais apropriadas. Pode-se ver, por exemplo, que a técnica de weight-decay não apresentou sucesso: quando o número de ciclos era grande, não conseguia aprender, desempenho pouco acima da escolha randômica; quando o número de ciclos não era grande, o sistema não convergia.

Como demonstrado na seção de resultados, o sistema proposto é promissor para lidar com este tipo de tarefa.

Referências

- [1] Andrade, G. A., Martins, W., "Inteligência Artificial Aplicada ao Acasalamento de Gado Nelore", Anais do VI Seminário de Pesquisa da UFG, Goiânia, GO, 1998, pp 92-93.
- [2] Vieira, M. A. C., Martins, W., "Abordagem Neuronebulosa para Desenvolvimento de Preditores", Anais do VI Seminário de Pesquisa da UFG, Goiânia, GO, 1998, pp 96-97.
- [3] Martins, W. et al, "Um Sistema Inteligente para o Acasalamento de Gado Nelore", Anais do V Simpósio Brasileiro de Redes Neurais, Belo Horizonte, MG, 1998, pp 315-318.
- [4] Bisant, D., Brown, D., "Neural Networks Models for Quantitative Genetics with Application to Dairy Cattle", Anais do World Congress on Neural Networks, 1994, vol. I, San Diego, EUA, pp. 297-301.
- [5] Haykin, S., *Neural Networks – A Comprehensive Foundation*, Mcmillan College Publishing Company, New York, 1995.
- [6] Franco, R. B., Relatório Interno LSI número 002/99, Escola de Engenharia Elétrica, Goiânia – GO.