

# TREINAMENTO DE REDES NEURAIS ARTIFICIAIS: O ALGORITMO BACKPROPAGATION

**Maria Carolina Stockler Barca<sup>1</sup>, Tiago Redondo de Siqueira Silveira, Marcio Magini<sup>2</sup>**

Universidade do Vale do Paraíba, Estrada Municipal do Limoeiro, 250 – CEP: 12305-810 – Jardim Dora – Jacareí - SP, <sup>1</sup>carolina\_stockler@hotmail.com, <sup>2</sup>magini@univap.br

**Resumo-** Este trabalho tem o intuito de mostrar o treinamento de uma Rede Neural Artificial, mais especificamente o algoritmo backpropagation. Para tanto, foi utilizada a ferramenta Neuframe<sup>®</sup> da Neuscience. Os estudos aqui propostos estão em caráter preliminar, porém mostram as diferentes topologias e a influência das diferentes funções de ativação no treinamento da rede. Também são definidas as taxas de aprendizado, “momentum”, erro e número de épocas. O estudo dessas taxas tem uma importância significativa quando, em caráter posterior, o algoritmo Backpropagation for utilizado em uma aplicação real.

**Palavras-chave:** Redes Neurais Artificiais, Inteligência Artificial.

**Área do Conhecimento:** 1 - Ciências Exatas e da Terra

## Introdução

A inteligência é a capacidade que um ser, humano ou não, possui para conhecer, aprender e representar. Dentre as características que um ser inteligente possui, o aprendizado é a mais importante, pois é a partir dele que se torna possível qualquer tomada de decisão, conclusão e desenvolvimento de criatividade. O aprendizado está relacionado com os conceitos de: adaptação, correção, otimização, interação e representação [9].

Dentro deste contexto, várias tentativas de imitar a inteligência humana através de computador foram feitas, dando origem ao que é chamado nos dias atuais de inteligência artificial (IA). Sabe-se que a IA tem como premissa copiar a inteligência humana. Assim, ela é definida como um conjunto de teorias e técnicas de implementação lógica que permite a uma máquina a capacidade de adquirir conhecimento [10]. A rede neural se insere neste contexto com uma dessas técnicas.

Para entender uma rede neural artificial (RNA), é necessário fazer um paralelo entre um neurônio biológico e artificial. O neurônio biológico é formado pelo soma que é o corpo do neurônio, pelo axônio que conduz o impulso nervoso de um neurônio para outro, por dendritos que são ramificações que fazem a comunicação entre um neurônio e as sinapses, que por sua vez propagam o sinal neural ativando a comunicação com o próximo neurônio [11].

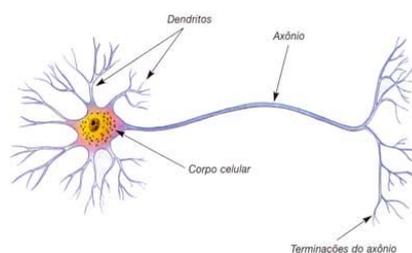


Figura1 – Neurônio Biológico

No neurônio artificial os dendritos são representados pelos valores de entrada, os pesos são as sinapses, a função de soma faz o papel do corpo do neurônio e a função de transferência representa o axônio [4,7].



Figura 2 – Neurônio Artificial

É através de um conjunto de neurônios artificiais que uma RNA se torna capaz de aprender a realizar uma determinada tarefa, essa capacidade é obtida através do treinamento da rede com relação ao problema de interesse.

Para o treinamento de uma RNA, pode-se utilizar 3 tipos de aprendizado: o *supervisionado* onde é fornecida a resposta desejada e são adaptados os pesos; o *semi-supervisionado* onde a rede aprende através de resultados do tipo sucesso e insucesso, mesmo informando o resultado esperado a rede não tem um parâmetro de como proceder com esses resultados; outro tipo de aprendizado é o *não-supervisionado* onde somente os dados de entrada são fornecidos e a

rede consegue se rearranjar até obter uma saída satisfatória [8,9]. As redes neurais são definidas através de algoritmos bem estabelecidos na literatura, conhecidos como Algoritmos de Aprendizado, que são definidos como procedimentos ou conjuntos pré-estabelecidos de regras. Basicamente, estes algoritmos diferem entre si pela forma de como é formulado o ajuste de um peso sináptico de um neurônio [2,12].

Por não existir um único algoritmo e cada um apresentar vantagens e desvantagens quanto à este ajuste sináptico, a escolha de um algoritmo de aprendizagem particular é influenciada pela tarefa de aprendizagem que uma rede neural deve executar.

Para uma tarefa onde o problema não é linearmente separável, é necessária a utilização de uma rede de múltiplas camadas [1]. O algoritmo mais popular e amplamente usado para este tipo de RNA é o Backpropagation [6].

Este algoritmo utiliza o aprendizado supervisionado, onde o processo de treinamento é feito com base em tentativa e erro, por isso, um dos problemas do algoritmo é seu tempo de treinamento e o que influencia nesse tempo é a taxa de atualização dos pesos, se a taxa definida for muito baixa a rede consome um tempo muito grande para o treinamento, por outro lado se a taxa é alta a rede consegue convergir em um pequeno espaço de tempo, porém, quando é apresentada uma outra entrada a rede se torna instável ocasionado outro problema que é a confiabilidade dos resultados [5].

Para minimizar o problema com o tempo de convergência deste algoritmo, durante o seu treinamento, existem algumas técnicas de otimização numérica, por padrão utiliza-se a técnica do gradiente descendente e da regra delta [6], porém, apesar de apresentarem uma complexidade maior, a técnica de Levenberg-Marquardt e a de Filtragem de Kalman têm demonstrado ser de grande eficiência quando comparadas à essas técnicas.

## Materiais e Métodos

Com base na metodologia de Kevin, são descritos alguns passos para treinar uma rede neural usando o algoritmo Backpropagation. O primeiro passo é a definição da arquitetura da rede e a inicialização dos pesos de cada nó das camadas [3,6].

Definida a topologia da rede, é iniciada a fase forward, utilizada para definir a saída da rede para um dado padrão de entrada. Nesta fase, um padrão de atividade (vetor de entrada) é aplicado aos neurônios da rede e seu efeito se propaga através da rede, camada por camada, até que uma saída seja gerada pela camada de saída. Então esta saída é comparada à saída desejada, e

um sinal de erro é calculado para cada nó de saída [3,6].

Sequencialmente a fase backward utiliza a saída desejada e a saída fornecida pela rede para atualizar os pesos de suas conexões, é feita uma transmissão dos sinais de erro calculados anteriormente em direção contrária, da camada de saída para cada nó nas camadas intermediárias, contra a direção das conexões. Os pesos sinápticos são ajustados, baseados nos sinais dos erros, para fazer com que a resposta real da rede se mova para mais perto da resposta desejada, em um sentido estatístico. Este processo continua até que o sinal de saída seja equivalente ao sinal de entrada ou se atingir um erro médio desejado [3,6].

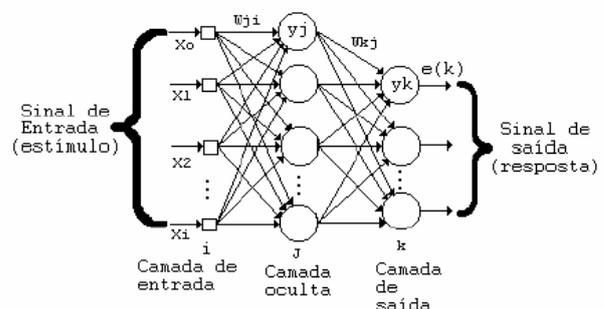


Figura 3 – Esquema de uma Rede Multi-Camadas

## Resultados Preliminares

Para obtenção dos resultados preliminares, foi utilizada uma base de testes que através de medidas das pétalas de alguns tipos de flores, após o treinamento da rede foi possível gerar uma probabilidade para classificação da flor pelas medidas informadas.

Na configuração da rede foram definidas algumas taxas:

- Taxa de Aprendizado: 0,2
- Taxa Momentum: 0,8
- Critério de parada: Taxa de Erro < 0,05
- Taxa de Erro de Teste < 0,2

Main Training Targets Training Inputs					
Main Sheet:Area [1.1]-[150,92] Setosa					
All	True Type	Petal Length	Petal Width	Sepal Length	Sepal Width
1	Setosa	5,00	4,00	1,00	0,00
2	Setosa	4,00	3,00	1,00	0,00
3	Setosa	4,00	3,00	1,00	0,00
4	Setosa	4,00	3,00	1,00	0,00
5	Setosa	5,00	4,00	1,00	0,00
6	Setosa	5,00	3,00	1,00	0,00
7	Setosa	4,00	3,00	1,00	0,00
8	Setosa	4,00	3,00	1,00	0,00
9	Setosa	5,00	3,00	1,00	0,00
10	Setosa	5,00	3,00	1,00	0,00
11	Versicola	5,00	2,00	4,00	1,00

Figura 4 – Dados de Entrada

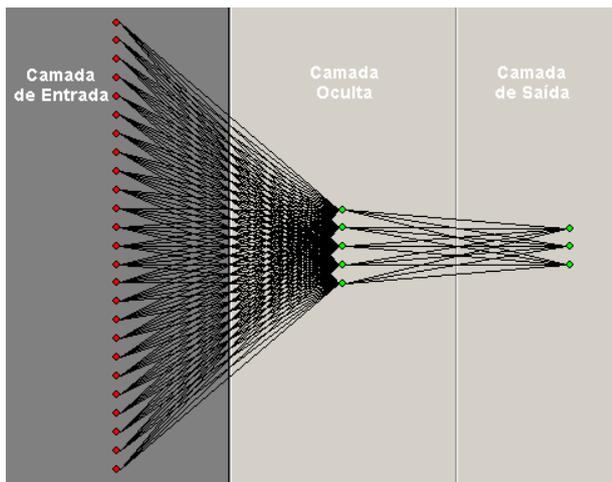


Figura 5 – Topologia definida para a base de teste

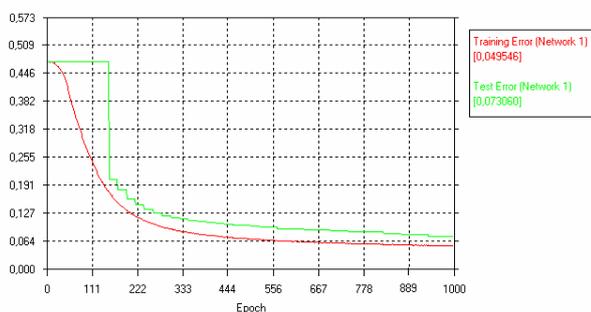


Figura 6 – Gráfico de Erro durante o treinamento

Main Network Results Known Results				
Main Sheet:Area [1,1]-(150,50) Virginia				
All	A	B	C	D
1	Virginia	95,61%	Virginia	
2	Setosa	97,36%	Setosa	
3	Virginia	97,91%	Virginia	
4	Versicola	79,67%	Versicola	
5	Versicola	93,13%	Versicola	
6	Setosa	97,64%	Setosa	
7	Setosa	95,55%	Setosa	
8	Setosa	92,79%	Setosa	
9	Versicola	91,11%	Versicola	
10	Setosa	96%	Setosa	
11	Setosa	95,55%	Setosa	

Figura 7 - Resultados

## Discussão

Durante o treinamento preliminar e estudo do comportamento da rede foi possível observar o tempo de convergência, a taxa de erro atingida e a confiabilidade dos resultados.

Com o aumento da taxa de aprendizado independente do valor definido para o momentum a rede se tornou instável.

Por outro lado, diminuindo-se a taxa de aprendizado tem-se uma rede mais confiável. Em conjunto com esta taxa, se não for definido um momentum ou o mesmo for baixo, o tempo de convergência da rede se torna muito longo, porém se o momentum definido for relativamente alto – proporcional à taxa de aprendizado – o tempo de convergência decresce visivelmente.

Outro parâmetro que infere no aumento do tempo de treinamento da rede é número de nós que a camada oculta possui.

Porém, independente da parametrização da rede, a definição da função de ativação da camada oculta também influencia no tempo de convergência e estabilidade da rede. Se a função de ativação usada for a sigmóide tem-se no final do treinamento uma rede mais estável, contudo o tempo para o mesmo é maior. Com a função de ativação tangente obtêm-se resultados proporcionalmente inversos.

## Conclusão

Os testes preliminares mostraram que as diferentes funções podem gerar resultados de treinamentos distintos. Não obstante, a implementação de forma consistente do algoritmo backpropagation depende fortemente dos parâmetros analisados aqui. Em perspectiva futura tem-se a intenção de implementar este algoritmo, já se fixando uma função de treinamento, para um problema real.

## Referências

- [1] CYBENKO, G. Approximation by superpositions of a sigmoid function. Mathematics of Control, Signals and Systems, 1989, p. 303-314.
- [2] HAYKIN, S. Neural Networks a Comprehensive Foundation. Second Edition, Prentice Hall, 1999.
- [3] KNIGHT, K.. Connection Ideas and Algorithms. Communications of the ACM., November 1990/Vol.33, No.11.
- [4] MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. Bulletin of Mathematical Biophysics, 1943.
- [5] MINSKY, M.; PAPERT, S. Perceptrons: An introduction to computational geometry. MIT Press, Massachusetts, 1969.
- [6] RUMMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. Internal Representation by Error

Propagation, in Parallel Distributed Processing. MIT Press, Cambridge, 1986.

[7] ROSENBLATT, F. The perceptron: A probabilistic model for information storage and organization in the brain, 1958.

[8] BLUMENAU (Santa Catarina). Universidade Regional de Blumenau. In: PROTÓTIPO DE UM SISTEMA DE IDENTIFICAÇÃO DE MINÚCIAS EM IMPRESSÕES DIGITAIS UTILIZANDO REDES NEURAS ARTIFICIAIS FEEDFORWARD MULTICAMADA. Blumenau, 2002. Disponível em: <<http://www.inf.furb.br/~jomi/tccs/2002-1/rafael/TCC2002-1-61-VF-RafaelGumz.pdf>>. Acesso em: 10 abr. 2005 – 10h11

[9] OSÓRIO, F. S.; BITTENCOURT, J. R. Sistemas Inteligentes Baseados em Redes Neurais Artificiais Aplicados ao Processamento de Imagens. In: I WORKSHOP DE INTELIGÊNCIA ARTIFICIAL, 2000, Santa Cruz do Sul. 2000. Disponível em: <<http://www.inf.unisinos.br/~jrbitt/publica.htm>>. Acesso em: 04 jan. 2005 – 11h23.

[10] LAROUSSE. Grande Enciclopédia Larousse Cultural. Editora Nova Cultural, 1999.

[11] LOESCH, Cláudio. Redes neurais artificiais, fundamentos e modelos. Blumenau: Editora da FURB, 1996, p. 166.

[12] MENDEL, J. M.; MCLAREN, R. W. Adaptive, Learning and Pattern Recognition Systems; Theory and Applications, chapter Reinforcement-learning control and pattern recognition systems. New York: Academic Press, 70.