

**UNIVERSIDADE FEDERAL DO PIAUÍ – UFPI  
CENTRO DE CIÊNCIAS DA NATUREZA – CCN**

**Francisco Neto Carvalho de Araújo  
Hugo Santos Piauilino Neto  
Luís Guilherme Teixeira dos Santos**

**IMPLEMENTAÇÃO DE UM PERCEPTRON DE MÚLTIPLAS CAMADAS PARA  
IDENTIFICAÇÃO DE TIPOS DE VIDROS**

**Teresina, 2015**

**Francisco Neto Carvalho de Araújo**  
**Hugo Santos Piauilino Neto**  
**Luís Guilherme Teixeira dos Santos**

**IMPLEMENTAÇÃO DE UM PERCEPTRON DE MÚLTIPLAS CAMADAS PARA  
IDENTIFICAÇÃO DE TIPOS DE VIDROS**

Relatório técnico apresentado como  
requisito para obtenção de nota referente  
ao segundo trabalho prático da disciplina  
Tópicos em Inteligência Artificial.

**Teresina, 2015**

## INTRODUÇÃO

As Redes Neurais Artificiais são modelos matemáticos inspirados pelos neurônios biológicos constituintes da estrutura cerebral. Os modelos neurais, ou sistemas conexionistas, afirma que a inteligência surge por meio de um processo de aprendizado, ou de adaptação, pelo qual as conexões entre os componentes (neurônios biológicos ou artificiais) são ajustadas (Luger, 2013). O processamento em sistemas como esse é distribuído ao longo de um conjunto de camadas de neurônios que processam as suas entradas de forma simultânea e independente.

Mitchell (1997) descreve a Rede Neural como um método que fornece uma abordagem robusta para aproximação entre o valor avaliado e o valor real, sendo atualmente o método mais efetivo de aprendizado para interpretação de dados complexos.

Os algoritmos e as arquiteturas que implementam essas técnicas são normalmente treinados, em vez de serem explicitamente programados, sendo este o grande poder da abordagem. Uma arquitetura de rede e um algoritmo de aprendizado, projetados adequadamente, podem capturar invariâncias do mundo sem serem explicitamente programados para reconhecê-las.

Luger (2013) destaca tarefas para as quais a abordagem neural é adequada, a saber:

*classificação*: decide a categoria ou grupo ao qual pertence um valor de entrada;

*reconhecimento de padrões*: identifica a estrutura ou padrões nos dados;

*evocação de memória*: inclui o problema da memória endereçável por conteúdo;

*predição*: identifica, por exemplo, doenças a partir de sintomas, causas a partir de efeitos;

*otimização*: encontra a “melhor” organização de restrições;

*filtragem de ruído*: separa o sinal do ruído de fundo, retirando os componentes irrelevantes de um sinal.

O poder das Redes Neurais Artificiais está na sua capacidade de aprendizado. Em um perceptron Multicamadas, os neurônios estão conectados em camadas, onde os neurônios da camada  $n$  passam as suas ativações (saídas) para os neurônios da camada  $n+1$ . Isso significa que os erros no interior da rede podem se espalhar e evoluir em formas complexas e imprevistas por meio das camadas

sucessivas, tornando a análise da fonte de erro na camada de saída complexa.

O algoritmo *backpropagation* (retropropagação) fornece uma maneira de atribuir a cada neurônio a sua parcela de contribuição pelo erro da rede, permitindo ajustar os pesos de forma correspondente. A abordagem adotada pelo algoritmo consiste em iniciar na camada de saída e propagar o erro retroativamente através das camadas ocultas.

## DESENVOLVIMENTO

Neste trabalho visamos implementar uma Rede Neural Artificial do tipo Perceptron Multicamada para identificação de determinados tipos de vidro, identificados na especificação como A, B e C, dependentes de quatro variáveis que representam a constituição química do tipos de vidro: Bário ( $x_1$ ), Potássio ( $x_2$ ), Silício ( $x_3$ ) e Cálcio ( $x_4$ ). As saídas da rede são mapeadas para cada tipo de vidro de acordo com a Tabela 1.

**Tabela 1: Mapeamento do tipo de vidro para saídas da rede.**

Conservante	$Y_1$	$Y_2$	$Y_3$
A	1	0	0
B	0	1	0
C	0	0	1

### Objetivo geral

- Implementar uma Rede Neural Artificial.

### Objetivos específicos

- Implementar um Perceptron Multicamadas;
- Testar o Perceptron na identificação de tipos de vidro;
- Implementar uma interface gráfica com o usuário;
- Analisar os resultados.

## PROCEDIMENTOS EXPERIMENTAIS

## Materiais

Na implementação foi utilizada a linguagem de programação Python na versão 3.x.x, adicionando-se as bibliotecas Numpy, para manipulação de matrizes, e Graphviz, para gerar o diagrama com a arquitetura da rede, os pesos finais de cada aresta e as saídas finais de cada neurônio da camada de saída da rede. Os computadores utilizados para implementação e testes utilizavam sistemas operacionais Unix (Ubuntu e Mac OS X).

## Métodos

A arquitetura a ser utilizada foi definida na especificação do trabalho e contém uma camada com quatro entradas, uma camada oculta com quatro neurônios e uma camada com três neurônios na saída, como pode ser vista na Figura 1. Os pesos iniciais são definidos inicialmente com valores aleatórios.

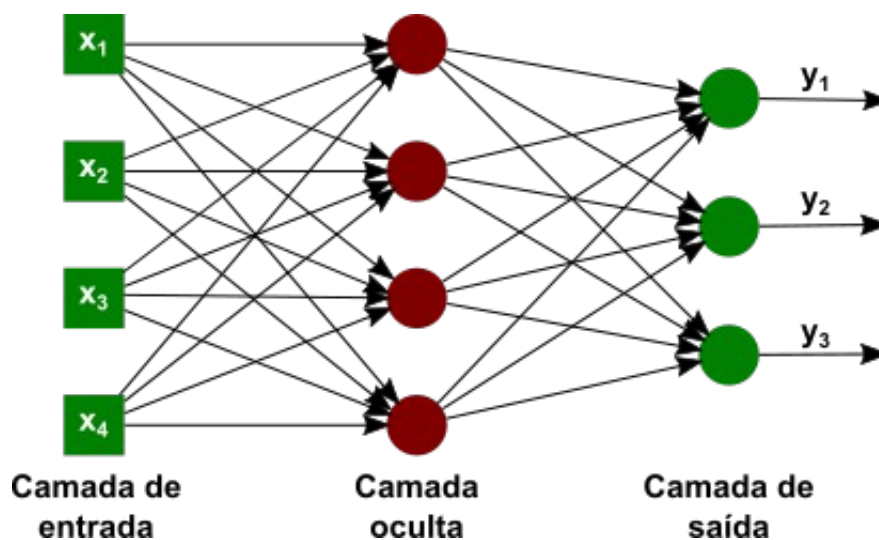
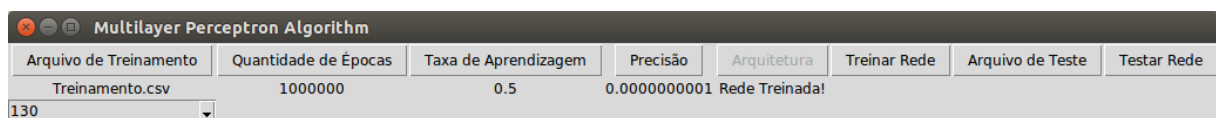


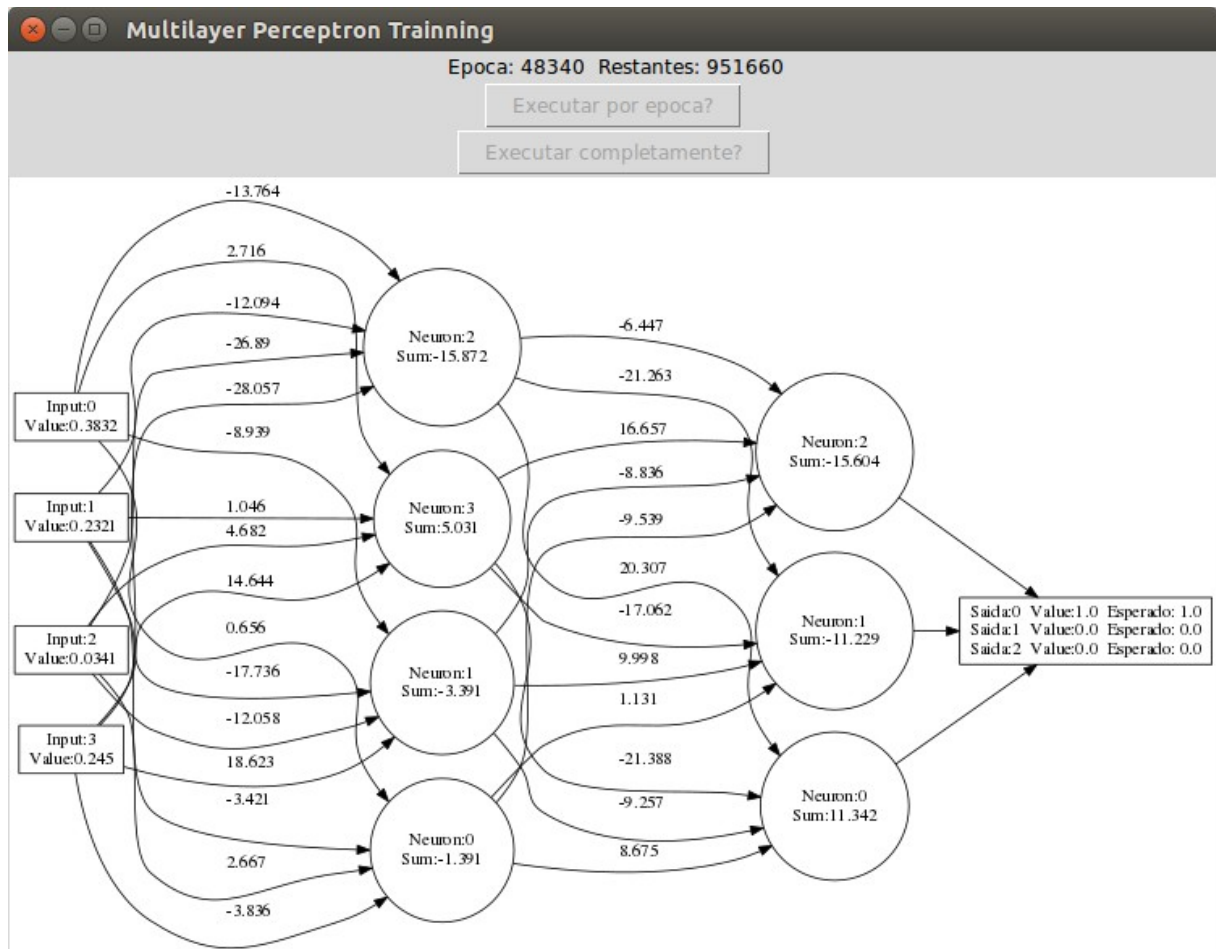
Figura 1: Arquitetura da Rede.

Para interação com o usuário foi implementada uma interface gráfica que pode ser vista na Figura 2. A interface permite a obtenção de arquivos de treinamento e testes, fornecidos através de arquivos CSV (*Coma Separated Value*), a definição da quantidade de épocas, da taxa de aprendizagem, bem como iniciar os processos de treinamento e teste da rede.



**Figura 2: Tela de configurações.**

A Figura 3 mostra a tela com a configuração final da rede obtida após o treinamento.



**Figura 3: Tela de exibição do treinamento.**

## RESULTADOS E DISCUSSÕES

O processo de treinamento da rede foi realizado utilizando toda a base de treinamento, ou seja, as 130 entradas foram apresentadas para a rede a cada época. As precisões (erro aceitável) utilizadas foram  $1 \times 10^{-7}$ ,  $1 \times 10^{-8}$  e  $1 \times 10^{-9}$ . A Tabela 2 mostra o tempo de execução e o número de épocas necessárias para a conclusão do treinamento. Em todos os casos a taxa de aprendizado foi mantida com 0,5.

Tabela 2: Precisão, tempo de execução e épocas de treinamento para os casos testados.

Precisão	Tempo de execução (s)	Épocas de treinamento
$1 \times 10^{-7}$	112,45	4333
$1 \times 10^{-8}$	164,78	6402
$1 \times 10^{-9}$	217,62	8395

Nas Figuras 4, 5 e 6 temos os gráficos para cada um dos casos citados anteriormente. Neles é possível ver claramente a redução do erro quadrático médio com o passar das épocas.

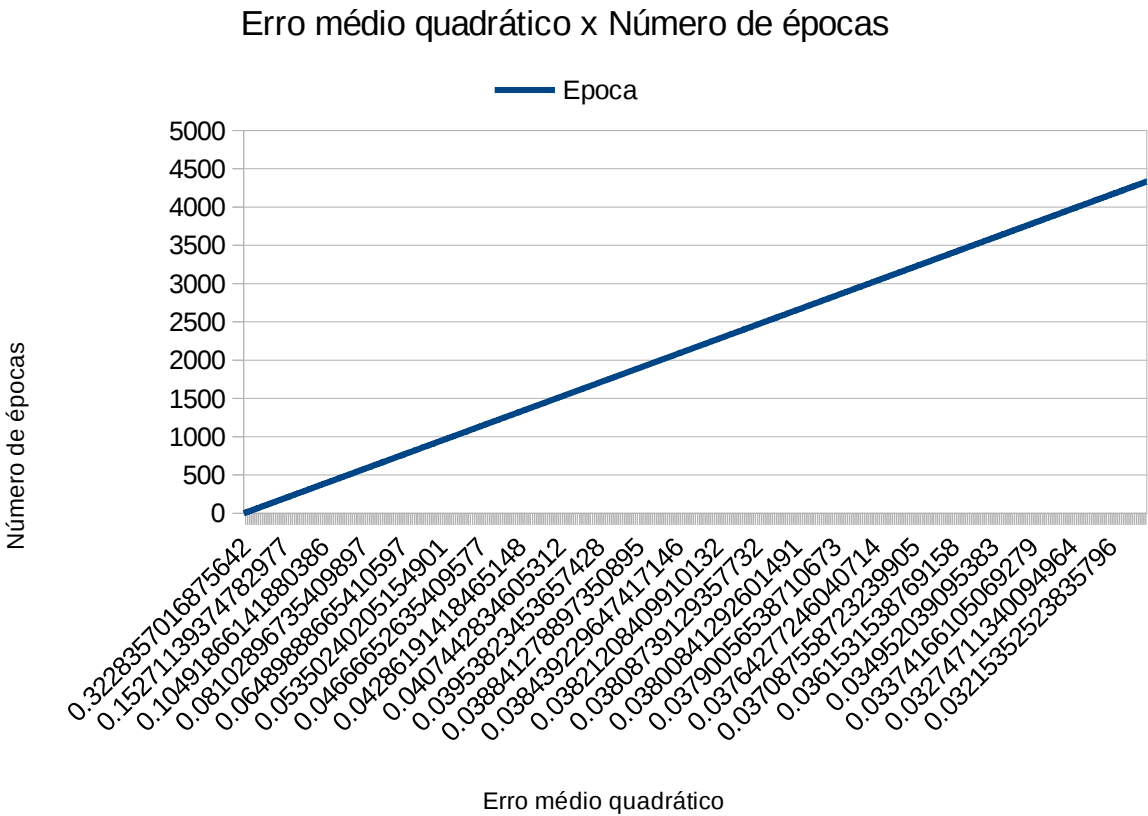
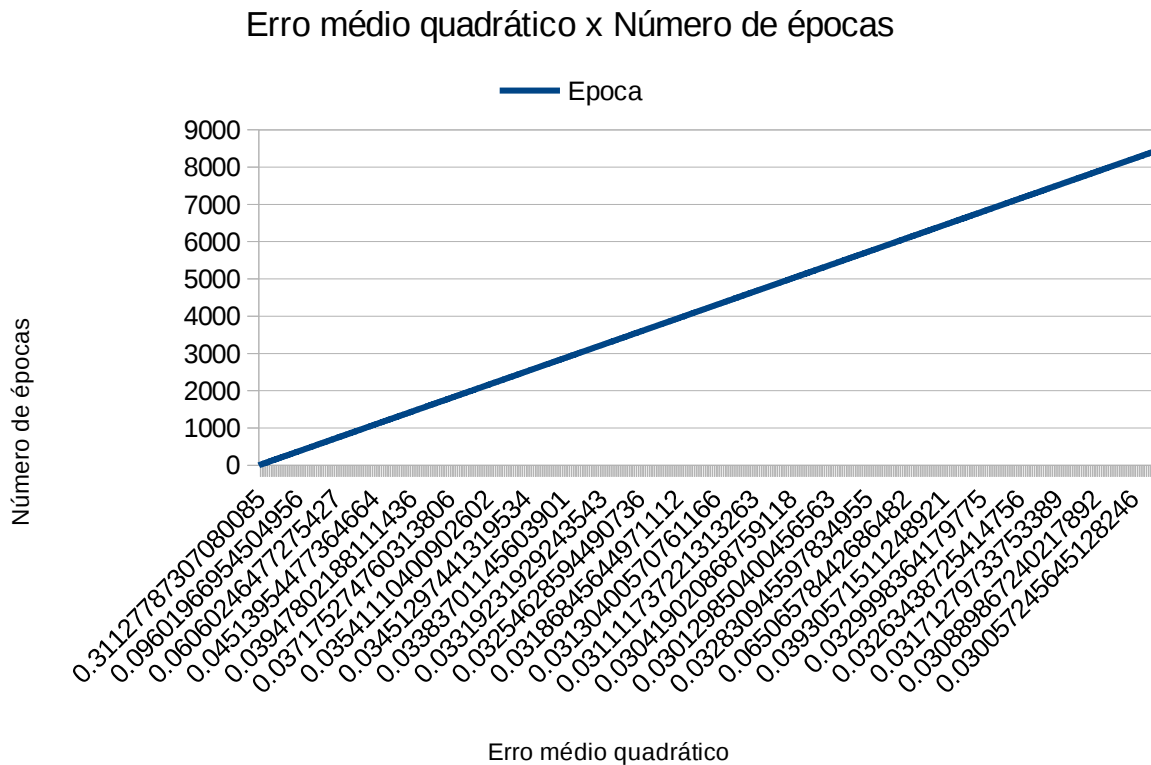


Figura 4: Relação entre o número de épocas e o erro quadrático médio para uma precisão de  $1 \times 10^{-7}$ .



**Figura 5: Relação entre o número de épocas e o erro quadrático médio para uma precisão de  $1 \times 10^{-8}$ .**





**Figura 6: Relação entre o número de épocas e o erro quadrático médio para uma precisão de  $1 \times 10^{-9}$ .**

## CONCLUSÃO

Quando submetida às entradas de teste, após realizado o treinamento, as rede conseguiu classificar corretamente, todas as amostras, com pequenas variações na precisão. Como era de se esperar, o erro médio quadrático reduz com o passar das épocas, fazendo com que o valor de saída convirja para o valor esperado. Outra confirmação é o aumento do número de épocas necessário para o sucesso do treinamento do Perceptron Multicamada, quando se reduz o erro aceitável. Assim os dados comprovam a eficiência da rede implementada para a classificação das amostras.

## REFERÊNCIAS

LUGER, G. F. *Inteligência artificial*. Tradução: Daniel Vieira. 6 ed. São Paulo: Pearson Education do Brasil, 2013.

MITCHELL, T. M. *Machine learning*, 1997.