

# RECONHECIMENTO DE VOGAIS UTILIZANDO REDES NEURAIS ARTIFICIAIS

Sóstenes Gonçalves Souza<sup>1</sup>, Profa: Mestra Cristina Maria Valadares de Lima<sup>1</sup>,  
Fabiano de Oliveira Rates<sup>2</sup>

<sup>1</sup>Engenharia da Computação – Universidade do Estado de Minas Gerais(UEMG)  
Avenida Paraná, 3001, Jardim Belvedere, 35501-170, Divinópolis, Minas Gerais, Brasil.

<sup>2</sup>Inovação Computação Móvel  
Avenida Paraná, 1348, Sidil, 35501-660, Divinópolis, Minas Gerais, Brasil.

sostenes198br@hotmail.com, crismvaladares@gmail.com, forates8@hotmail.com

**Abstract.** *Current handwriting recognition techniques, used as methods of inputting information in most portable computers, have brought a number of problems of accuracy and comfort to users, since it determines a proper writing pattern that should be used by the user. This paper describes the development and development of an artificial neural network modeled for the recognition of vowels of distinct users, the present training algorithm used for the training of the artificial neural network was the algorithm backpropagation, in order to recognize calligraphy of different users. The results obtained with a hit rate of over 75% with calligraphy of different users.*

**Resumo.** *As atuais técnicas de reconhecimento de escrita, usadas como métodos de entrada de informações em grande parte dos computadores portáteis, trouxeram diversos problemas de precisão e conforto para os usuários, uma vez que determina um padrão próprio de escrita que deve ser utilizado pelo usuário. Este artigo descreve a elaboração e desenvolvimento de uma rede neural artificial modelada para o reconhecimento de vogais de usuários distintos, o presente algoritmo de treinamento utilizado para o treinamento da rede neural artificial foi o algoritmo backpropagation, visando reconhecer caligrafias de usuários distintos. Os resultados obtidos apresentaram ótimos resultados, com uma taxa de acerto de acerto superior a 75% com caligrafias de diversos usuários distintos.*

## 1. INTRODUÇÃO

A muito tempo as interfaces entre homens e máquinas, tem sido um constante problema que alimenta diversas linhas de pesquisas, principalmente em busca de novas técnicas de interação entre homens e máquinas. Ao longo das últimas décadas as interfaces de máquinas evoluíram, ao ponto de suportar recursos mais naturais, como a fala e a escrita. Ainda assim estes elementos ainda não se apresentaram satisfatoriamente mais eficientes que os convencionais, contudo estes elementos já têm demonstrado algumas capacidades, como suportar tarefas de autenticação, auxílio em desenhos, entre outros.

Com a popularização dos *SmartPhones* (celulares inteligentes), *SmartWatch* (relógios inteligentes), computadores portáteis, entre outros, o uso de teclados como

forma de entrada de dados, tem se tornado cada vez mais obsoleto. Com isto, viu-se a ampliação de estudos de técnicas para reconhecimento de escritas, visto que normalmente as técnicas adotadas para este tipo de interface trazem sérios problemas, como falta de precisão no reconhecimento da caligrafia do usuário e desconforto para o usuário, que normalmente tem que aprender um novo padrão alfabeto.

Dentre as técnicas existentes para resolver este problema, tem se destacado as Redes Neurais Artificiais (RNAs), como sistemas de reconhecimento de padrões desenvolvida para estresir a capacidade humana de ler textos, devido ao seu poder de aprendizagem e por sua fácil aplicação. Que por sua vez, são elementos interconectados que procura imitar a concepção do funcionamento do cérebro humano, com as RNAs voltadas ao reconhecimento de caracteres, busca-se primeiro o aprendizado, para posteriormente reconhecer os padrões que são provenientes da digitalização de imagens de texto. E por sua vez estes sinais são armazenados como forma de aprendizado, para posteriores consultas dos sinais.

O mais elementar algoritmo para o treinamento da RNA é o algoritmo *backpropagation*, ou em português retropropagação, é um método baseado na utilização de gradientes descendentes se baseando na regra delta, também conhecida como regra delta generalizada (OLVEIRA, 2013), sendo utilizado junto com a função Sigmoide, é uma função matemática, que visa aumentar o poder de reconhecimento da RNA, junto com o *framework* Encog que é um *framework* avançado de estrutura para aprendizagem de máquinas suportando uma variedade de algoritmos avançados, como o algoritmo *backpropagation* (HEATON, 2015).

Nesse sentido as RNAs em geral são sistemas computacionais, inspirados no sistema nervoso biológico (BRAGA; CARVALHO; LUDEMIR, 2000), com processadores maciço e paralelamente distribuído, constituído de unidades de processamento simples (neurônios), que têm a propensão natural para armazenar conhecimento experimental, tornado o disponível para uso posterior.

Tendo em vista esta tendência, (CRUZ; NEHME, 1993), abordam em seu artigo que os sistemas utilizando redes neurais para reconhecimento de caracteres são sistemas desenvolvidos para reproduzir a capacidade humana de ler textos, com o objetivo de construir uma RNA para o reconhecimento de caractere, utilizando técnicas avançadas para o processamento de imagens, visando aperfeiçoar o treinamento e posteriormente o reconhecimento da rede neural. E para o treinamento da rede neural foi comparado dois métodos de treinamentos nos quais são eles o modelo BAM (*bidirectional associative memory*) e o modelo *backpropagation*. Concluindo que as RNA podem obter o reconhecimento de caracteres de forma eficaz, com rapidez e segurança, podendo ser comparável à flexibilidade humana.

## 2. REFERENCIAL TEÓRICO

Esta seção descreve os principais conceitos demonstrados no presente trabalho, tais como reconhecimento de padrões, abordando mais detalhadamente o reconhecimento de caracteres, conceitos de uma rede neural, desde seus conceitos e estrutura até o aprendizado de máquinas.

## **2.1. RECONHECIMENTO DE PADRÕES**

O objetivo da utilização do reconhecimento de padrões é detectar, extrair e identificar elementos existentes em um cenário (conjunto de objetos, pertencentes ao mundo real, que podem ser percebidos a olho nu). Desta forma, procura-se imitar as habilidades dos seres humanos, criando dispositivos e algoritmos capazes de realizar estas habilidades humanas como classificar padrões, localizar objetos e detectar as possíveis relações entre estes elementos.

A principal vantagem é a de capacitar máquinas para realizar tarefas até então possíveis de serem realizadas somente por humanos. Suas principais aplicações são: computação de dados, aumento da capacidade de aquisição de dados pelo computador, sensoriamento remoto, controle de qualidade e classificações, além de diversas aplicações científicas e comerciais.

O reconhecimento de caracteres tem como objetivo relacionar uma entrada a um caractere pertencente ao alfabeto. Um único caractere de um alfabeto está sujeito a diversas variações quando é manuscrita. Porém mesmo com diversas variações, podem ser facilmente reconhecidas por humanos. Evidentemente estas formas abstratas não é apenas uma imagem, mas sim um conjunto de padrões que a representam (DIMAURO et al., 1997).

Sendo assim o reconhecimento de padrões é largamente usados em aplicações como o reconhecimento de fala, classificação de documentos em categorias, reconhecimento de escrita e reconhecimento de faces.

## **2.2. APRENDIZADO DE MÁQUINAS**

Aprendizado está diretamente interligado com a capacidade de compreender e resolver novos problemas e conflitos e de adaptar-se a novas situações, pois se um sistema é capaz de aprender a exercer uma determinada tarefa, pode-se então ser denominado de inteligente como é afirmado por Copin (2010). Segundo Santos (2005) e Mitchell (1997), o aprendizado de máquina nada mais é do que o aprendizado por experiência, que conforme são executadas as tarefas, o sistema vai se adequando para melhor forma de resolver o problema.

Existem diversos métodos para o aprendizado de máquina, entre eles o aprendizado por hábitos conforme citado por (SILVA; VANDERLINDE, 1993) e (COPIN, 2010). A abordagem utilizada para o aprendizado de máquina foi a abordagem de rede neural artificial para realizar o aprendizado de máquinas, assim desenvolvendo uma inteligência artificial para o reconhecimento de vogais.

## **2.3. REDES NEURAIS ARTIFICIAIS**

RNAs no geral são “máquinas” feitas para trabalharem segundo o mesmo processo de um sistema nervoso biológico. Seus elementos de processamento são denominados neurônios artificiais, interconectados, que efetuam a soma de entrada, gerando uma saída através da aplicação desta soma por uma função de transferência. Na prática são elementos computacionais de lógica não linear operando em paralelo, e dispostos de forma a representar uma rede neural biológica (LIPPMANN, 1987). A principal vantagem de representa RNAs, como o cérebro, está na capacidade de trabalhar sobre exemplos já existentes, dispensando a programação algorítmica (CRUZ; NEHME, 1993).

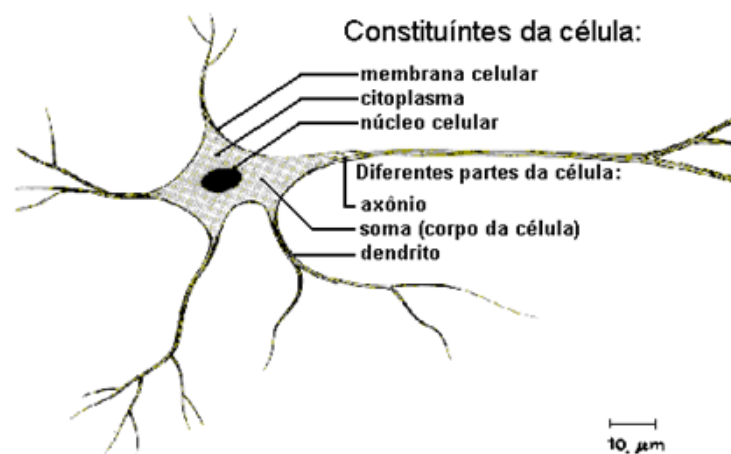
Assim, o cérebro, pode se adaptar ao que já existe, trabalhando como um sistema dinâmico não algorítmico (KOSKO, 1988), está falta de certeza é a principal vantagem da implementação de uma RNA, pois há vários elementos concorrendo para dar a resposta a um determinado estímulo de entrada, ganhando ao final a resposta mais gerada, como uma votação, assim proporcionando a capacidade de dar respostas consistentes a diversos estímulos distintos de entrada (CRUZ; NEHME, 1993). É em cima deste princípio que surge a utilização das RNAs para o reconhecimento de padrões, mesmo que a amostra a reconhecer tenha grandes variações da amostra original.

Assim observa-se, que as RNAs têm um imenso potencial no que diz respeito à substituição das tarefas notavelmente humanas, como o reconhecimento de voz, imagens, análise do mercado financeiro e outros comportamentos sociais e basicamente sua estrutura é composta por uma coleção de neurônios (nodos), agrupados em camadas.

### 2.3.1. NEURÔNIOS BIOLÓGICOS

Neurônios biológicos são responsáveis por captar os estímulos nos seus dendritos e os transmitir pelos axônios para que este estímulo atinja outro neurônio ou célula. Essas captações dos estímulos são realizadas por receptores sensoriais especializados, que podem ser estes mecânicos, térmicos, nociceptores, químicos, entre outros (TAFNER; XEREZ; R., 1995).

Os neurônios são estimulados através das sinapses (conexões interligando a rede de neurônios), conforme pode ser observado na figura 1, que conectam os dendritos de um neurônio aos axônios de outro neurônio ou com o sistema nervoso central. Esses estímulos, na forma de sinais, são aplicados a um sensor limiar, responsável por definir qual o nível de energia mínima de entrada acima do qual o neurônio será ativo.



**Figura 1. Funcionamento da plataforma proposta.**

Fonte: <http://www.din.uem.br/ia/neurais/image13.gif>

Caso a soma dos sinais for maior que o nível de limiar, o neurônio envia energia através dos axônios, onde a energia é então transferida para outras sinapses, ou também realimenta a sinapse original (RODRIGO, 2000), mas se caso a soma dos sinais for menor que o limiar pré-definido, nada acontece.

As camadas de uma RNA é uma colação de neurônios com a mesma função, e normalmente são classificadas em:

- (a) Entrada, o processamento é realizado sobre valores numéricos, onde em alguns casos, cada valor correspondente a somente uma entrada. O pré-processamento é necessário quando existem entradas envolvendo atributos qualitativos ou imagens, para efetuar sua conversão para valores numéricos (TURBAN, 1990);
- (b) Saída, é onde encontra-se a solução para o problema proposto. Tal qual a camada de entrada, está só pode assumir valores numéricos (RODRIGO, 2000). Nos casos em que o pré-processamento é realizado previamente na camada de entrada, não é exigido na camada de saída, porém é comum encontrar a ocorrência de pré-processamento nas duas camadas;

Em relação as suas conexões, podem ser feitas as seguintes observações:

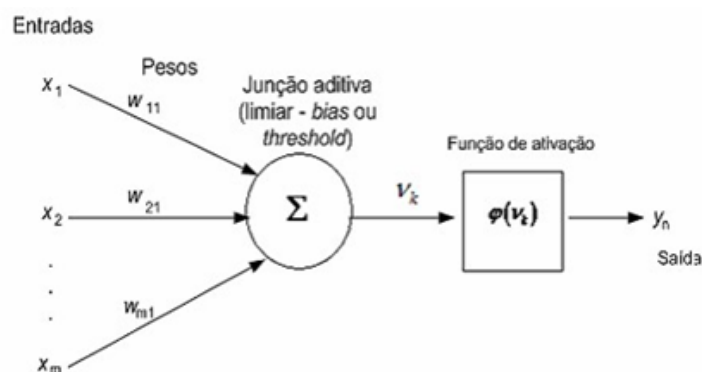
- (a) Peso das conexões são elementos fundamentais à rede. Segundo (FAUSSET, 1994) eles são responsáveis por exprimir a força relativa, em valor numérico, dos dados alimentados ou transmitidos às várias conexões;
- (b) Algumas conexões são, a conexão *feedback*, onde o resultado pode ser enviado para a camada posterior ou para camada anterior; a conexão *feedforward*, o resultado é somente enviado para a próxima camada; e a conexão *feedlateral*, onde o resultado é enviado para os elementos da mesma camada, devido a presença de conexões laterais;
- (c) A função de soma é responsável por efetuar os cálculos necessários para estabelecer a média dos pesos que compreendem as conexões dos neurônios da camada de entrada (TAFNER; XEREZ; R., 1995);
- (d) A função de transferência é responsável por analisar os resultados de entrada, verificando se o neurônio produziu ou não um resultado de saída. Com isso a função de transferência modifica o valor de saída do neurônio para que o mesmo respeite determinada faixa de domínio.

### 2.3.2. ELEMENTO DE PROCESSAMENTO

Um processo similar ao que ocorre no sistema nervoso central humano (cérebro) ocorre nos neurônios de uma RNA, também conhecido como elementos de processamento, como poder ser observado na figura 2, ou também são denominados de nodos.

Os sinais entram nos nodos através de conexões ponderadas, onde as informações estão armazenadas. Cada sinal que entra no sistema, seja este valor positivo ou negativo, é multiplicado por um número, ou peso que indica sua influência na saída do nodo (RODRIGO, 2000). Então é realizada a soma ponderada dos sinais, assim produzindo certo nível de atividade, como em um neurônio natural, uma resposta só é gerada num nodo se este nível de atividade ultrapassar um determinado limite. Caso não atinja esse limite, não haverá nenhuma saída, indicando que não há nenhuma informação a ser fornecida pelo nodo.

Existem dois tipos de RNAs: auto associativa, que correlaciona um padrão a ele mesmo, e hetero-associativo, que relaciona um padrão de entrada a um padrão de saída:



**Figura 2. Neurônio artificial.**

Fonte: <http://www.scielo.br/img/revistas/bcg/v16n3/a04fig03.jpg>

- (a) Rede auto associativa tem como principal característica a possibilidade de gerar um padrão completo, sem distorções de um padrão ruidoso. Cada tipo de rede auto associativas tem uma maneira diferente de padrão de aprendizado, e pode manter somente um certo número de padrões em sua memória;
- (b) Rede hetero-associativa segundo (TAFNER; XEREZ; R., 1995), uma rede hetero-associativa mapeia um conjunto de padrões (entrada) relacionados a outros (saídas). É típica para casos em que uma determinada entrada já esteja associada a uma determinada saída;

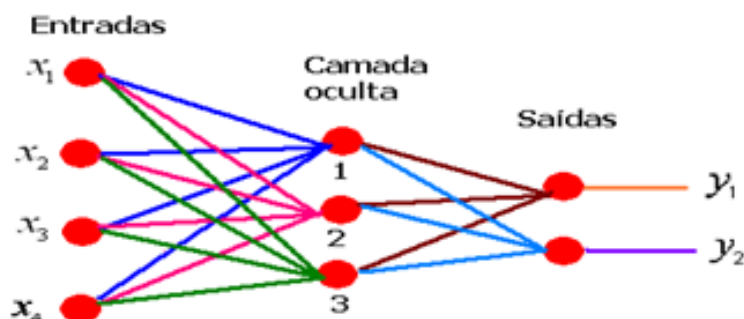
## 2.4. ALGORITMO DE TREINAMENTO *BACKPROPAGATION*

O treinamento é a atividade que leva a rede ao aprendizado. Denomina-se algoritmo de treinamento, um conjunto de regras bem definidas para a solução de um problema de aprendizado. Existem muitos tipos de algoritmos de treinamento, específicos para determinados modelos de RNAs, estes algoritmos diferem entre si principalmente pelo modo como os pesos são modificados durante as etapas do processo.

O algoritmo de treinamento *backpropagation* teve sua origem em meados de 1986, sendo um dos primeiros algoritmos criado para ser utilizado nas RNAs, este modelo é uma generalização dos métodos baseados no conceito que contém elementos computacionais individuais, formando uma camada que é conectada diretamente a saída (CRUZ; NEHME, 1993). Onde é, solicitada, no aprendizado para responder uma resposta pré-estabelecida. Neste modelo, há também camadas intermediárias não ligadas a saída, transformando-se em uma camada multinível (HINTON, 1989). “Desta forma, o erro – diferença entre a saída desejada e a atual – necessita ser retro propagado para as camadas intermediárias” (RUMELHART; HINTON; WILLIAMS, 1985).

Este algoritmo foi idealizado por Rumelhart, Hinton e Willians no ano de 1986, o termo surgiu do fato que o algoritmo se baseia na retro propagação dos erros para realizar os ajustes dos pesos das camadas intermediárias (RUMELHART; HINTON; WILLIAMS, 1985). A maneira de calcular as derivadas parciais do erro de saída em relação a cada um dos pesos da rede é o que caracteriza este algoritmo e seu treinamento ocorre em duas fases: um passo para frente, onde é denominado de propagação (*foward*), e um passo para trás, denominado de retropropagação (*backward*), conforme é demonstrado na figura 3, onde existem 4 neurônios de entrada ( $x_1, x_2, x_3, x_4$ ), onde cada um dos neurônios

se interconecta a todos os neurônios da camada intermediária (camada oculta) e cada neurônios da camada intermediária se interconecta a todos os neurônios da camada de saída.



**Figura 3. Algoritmo *Backpropagation*.**

Fonte: <http://www.lmtech.info/index.php/tecnologia/inteligencia-artificial/129-introducao-a-redes-neurais-artificiais>

Na propagação, o vetor de entrada é aplicado nos nós sensoriais da rede e seu efeito se propaga através da rede, camada a camada. Com isso um conjunto de saídas é produzido como a resposta real da RNA (OLVEIRA, 2013). No passo da retropropagação, os pesos do vetor de entrada são todos reajustados de acordo com as regras de correções de erros. Dessa forma, a resposta real da rede é subtraída de uma resposta desejada para produzir um sinal de erro (OLVEIRA, 2013). Este sinal de erro é então retropropagado para as camadas internas de entrada, fazendo com que a resposta real da rede se ajuste para o mais próximo da resposta desejada.

### 3. DESENVOLVIMENTO DO PROJETO

Esta seção descreve o fluxo de desenvolvimento do projeto, abordando o processamento de imagens, a obtenção dos dados e a forma de realizar o tratamento das imagens. Também descrevendo os parâmetros e configurações utilizados para construção da rede neural artificial.

#### 3.1. PROCESSAMENTO DE IMAGENS

Primeiramente foi definido o domínio na qual a RNA iria reconhecer, no qual este domínio são as vogais (A, E, I, O, U), conforme demonstrado na figura 4, demonstrando uma ilustração dos desenhos correspondentes as vogais a serem reconhecida pelas RNAs. As vogais foram escritas manualmente, durante o processo de treinamento, execução e validação das RNAs

A E I O U

**Figura 4. Formas manuscritas das vogais definidas – A, E, I, O, U.**

Fonte: **Elaborada pelo autor**

Cada forma manuscrita de uma vogal possui um código para identifica-lo, de forma a ser possível distinguir cada uma delas. Além disso, cada vogal possui também

**Tabela 1. Tabela com os valores mínimos, máximos e ideal de cada vogal e seus respectivos códigos**

Código	Vogal	Valor mínimo	Valor ideal	Valor máximo
1	O	0	0,1	0,19
2	I	0,2	0,3	0,39
3	A	0,4	0,5	0,59
4	E	0,6	0,7	0,79
5	U	0,8	0,9	0,99

uma faixa de valores pré-definidos, que começa em um valor mínimo, até um valor máximo, assim representando os possíveis retornos da rede neural artificial para cada vogal.

Caso o valor de saída da RNA, para uma vogal manuscrita, esteja dentro à faixa de valores mínimo e máximo definidos, significa que a RNA reconheceu a entrada correspondente a forma geométrica escrita. E o valor ideal para vogal foi definido, sendo a média entre o valor mínimo e o valor máximo, como é demonstrado na tabela 1, com o código de cada forma geométrica (vogal) e seus respectivos valores mínimos, máximo e ideal.

### **3.1.1. OBTENÇÃO DOS DADOS**

A obtenção dos dados foi realizada manualmente por um usuário, de forma que está pessoa escreveu uma vogal e também informou o código de qual vogal está sendo representada, de acordo com os códigos da tabela 1.

Pode-se observar através da figura 5, a interface disponibilizada para que o usuário possa entrar com dados de validação e treinamento da RNA. No exemplo em questão, é inserido uma vogal A com seu respectivo código de treinamento 3, em quanto na validação é inserido uma vogal O com seu respectivo código de treinamento 1, conforme a tabela 1.

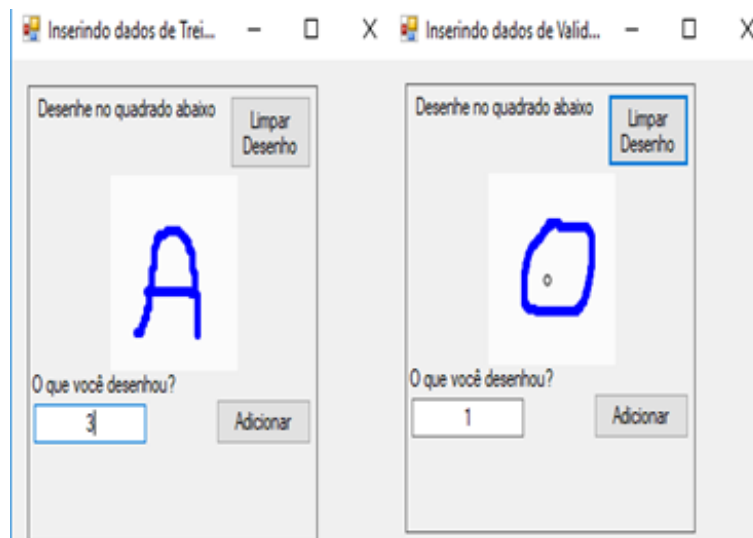
Porém apenas com a inserção destes dados, a rede neural artificial ainda não é capaz de prever qual a forma da imagem fornecida, pois, primeiro é necessário realizar o tratamento desses dados (imagem), fornecidos pelo usuário, visando facilitar o reconhecimento da RNA, como será demonstrado na próxima seção.

### **3.1.2. TRATAMENTO DAS IMAGENS**

Para que a RNA consiga receber os dados (imagens) recebidas pelos usuários, não basta simplesmente indicar que a entrada será tal imagem informada. Para isso, é necessário realiza um tratamento da imagem a ser inserida para o treinamento da RNA, pois se nenhum tratamento for realizado, a entrada não será bem aceita pela rede, assim aumentando o tempo de aprendizado e diminuindo sua taxa de acerto.

Primeiramente é realizado a inversão das cores para preto e branco, assim tornado possível a utilização no *framework* AForge. Após a inversão das cores é então realizada a normalização dos dados de entrada (imagem), pegando a imagem de entrada de resolução





**Figura 5. Exemplo de inserção de dados para a RNA.**

Fonte: **Elaborada pelo autor**

100x100 e dividindo à em 100 partes de 10x10 *pixels*, onde cada parte foi normalizada utilizando a seguinte equação

$$y(x) = \left(\frac{1}{x}\right),$$

onde:

- y é o valor normalizado;
- x é valor a ser normalizado.

Após realizada a divisão da imagem em 100 regiões, cada região 10x10 *pixels*, foi então realizada a normalização da “população” de bits de cada região, que nada mais é do que contabilizar os pontos pretos de cada parte. Essa quantidade de pontos pretos de cada região é o valor de “x” da formula de normalização descrita a cima, a partir desse valor normalizado, pode-se então gerar o valor de entrada para a RNA.

Através das normalizações realizadas nas entradas de dados (imagens) foi possível reduzir consideravelmente a quantidade de neurônios da camada de entrada, assim melhorando a eficácia dos treinamentos das RNAs. E para isso, foi utilizada a api gráfica AForge, realizando-se os passos. Primeiramente obtendo a imagem original, posteriormente invertendo a cor da imagem para preto e branco, assim com as cores invertidas o *framework* AForge consegue detectar os 4 extremos da imagem, assim através desses extremos o *framework* escala a imagem para o tamanho desejado (100x100), após realizado a expansão da imagem ela é então novamente invertida para o fundo branco e preto.

Após realizado a padronização dos dados (imagens) de entrada para a RNA, constatou-se que os resultados gerados foram bem mais eficientes. Dessa forma, a rede pôde aproveitar melhor as entradas, assim reduzindo a quantidade de neurônios na camada de entrada, reduzindo também a quantidade de exemplos necessários para o treinamento,

assim aumentando a qualidade e a diversidade de tamanhos possíveis para a predição da RNA.

### **3.2. REDE NEURAL ARTIFICIAL**

Nesta seção será descrita detalhadamente os parâmetros, definições e configurações utilizadas para a construção da rede neural artificial.

#### **3.2.1. FUNÇÃO DE ATIVAÇÃO**

Para a construção da RNA deste projeto foi utilizada a função de ativação sigmoide, pois a mesma é vantajosa quando utilizadas com algoritmos de retropropagação, pois ser fácil de distinguir, e ser capaz de minimizar a capacidade de computação para o treinamento. O termo sigmoide significa “forma de S”. Com isto esta função pode assumir todos os valores entre 0 e 1. Sendo assim, ela aceita toda as faixas de valores escolhidas para a representação das vogais, e esta função pode ser representada através da equação:

$$g(x) = \frac{1}{(1 + e^{-x})},$$

#### **3.2.2. NÚMERO DE NEURÔNIOS EM CADA CAMADA**

A camada de entrada da RNA proposta apresenta 100 neurônios, sendo um neurônio para cada *pixel*, assim representando cada um dos elementos da matriz 10x10 que representa a vogal manuscrita e normalizada de um usuário qualquer. Na camada de saída, a RNA possui apenas um neurônio, para retornar o valor predito que representa a imagem identificada pela RNA. E, por fim, para a definição do número de neurônios presentes na camada oculta, foram utilizadas duas configurações calculadas pelas regras descritas nos próximos tópicos:

#### **3.2.3. REGRA DO VALOR MÉDIO**

A regra do valor médio pode ser definida pela equação:

$$\frac{N1 + N2}{2}$$

Onde:

- N1 é o número de neurônios na camada de entrada;
- N2 é o número de neurônios na camada de saída.

#### **3.2.4. REGRA DA RAIZ QUADRADA**

A regra da raiz quadrada é definida pela equação:

$$\sqrt{N1 + N2},$$

Onde:

- N1 é o número de neurônios na camada de entrada;
- N2 é o número de neurônios na camada de saída.

### 3.2.5. DEFINIÇÕES DAS CONFIGURAÇÕES

Baseado nas informações citadas no tópico anterior, obteve-se as seguintes configurações para a elaboração das RNA. Utilizando a regra do valor médio, foi obtido a seguinte configuração: 100 neurônios na camada de entrada 50 neurônios na camada oculta e 1 neurônio na camada de saída. Utilizando a regra da raiz quadrada, foi obtida a seguinte configuração: 100 neurônios na camada de entrada, 10 neurônios na camada oculta e 1 neurônio na camada de saída.

Visando evitar qualquer falha na comparação destas configurações propostas, todas as RNAs de uma determinada configuração começam sempre com o mesmo peso inicial. Para armazenar os pesos resultantes de uma RNA após o treinamento, foram utilizados os próprios métodos do *framework* Encog, assim facilitando a reutilização desta rede sem ter que a treina-las novamente.

### 3.2.6. QUANTIDADE DE EXEMPLOS NO TREINAMENTO DE VALIDAÇÃO

Após definido as configurações das RNAs a serem utilizadas, é preciso definir o número de exemplos a serem utilizados durante o treinamento. Com o intuito de analisar como cada configuração se comporta com diferentes quantidades de exemplos de entrada, foram utilizadas as seguintes quantidades:

- 10 exemplos de cada vogal, totalizando 50 exemplos;
- 20 exemplos de cada vogal, totalizando 100 exemplos;
- 30 exemplos de cada vogal, totalizando 150 exemplos.

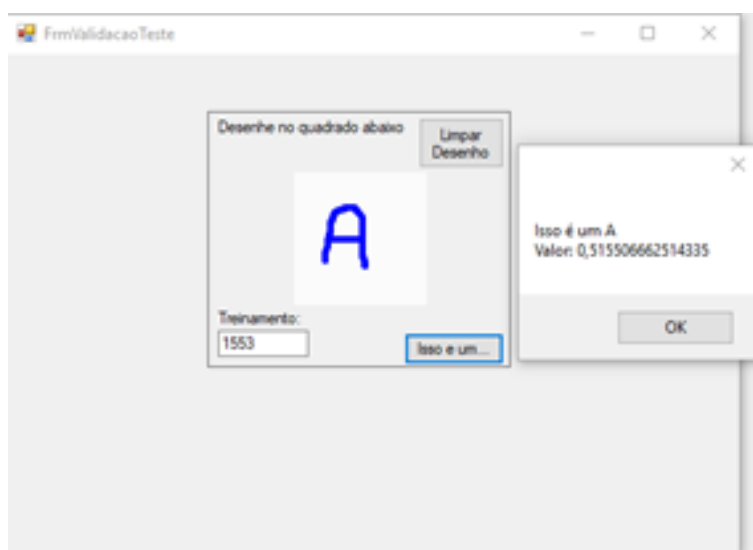
Para realizar a validação do treinamento da RNA foram criados mais 150 exemplos, sendo 30 exemplos de cada vogal, assim colocando em prova a capacidade da RNA distinguir padrões mesmo com um razoável volume de ruídos, apenas ressaltando que todos os exemplos foram desenhados manualmente por um usuário (utilizando o mouse), podendo apresentar, além de tamanhos variados, pequenas deformações inseridas propositalmente, para que a RNA não trabalhe apenas com formas perfeitas.

## 4. RESULTADOS OBTIDOS

Nesta seção são exibidos a tela para o reconhecimento das vogais, todos os resultados obtidos, divididos por configuração da RNA e pelo número de exemplos utilizados no treinamento.

### 4.1. TELA DE RECONHECIMENTO DE VOGAIS

A tela para o reconhecimento de vogais (Validação) é similar a tela para obtenção dos dados, onde o usuário irá desenhar uma vogal, informando também o código do treinamento, assim realizando a pesquisa e obtendo a resposta encontra pela rede neural artificial, como é demonstrado na figura 6



**Figura 6. Tela para o reconhecimento de vogais.**

Fonte: Elaborada pelo autor

### 4.2. CONFIGURAÇÃO 1 - UTILIZANDO A REGRA DO VALOR MÉDIO

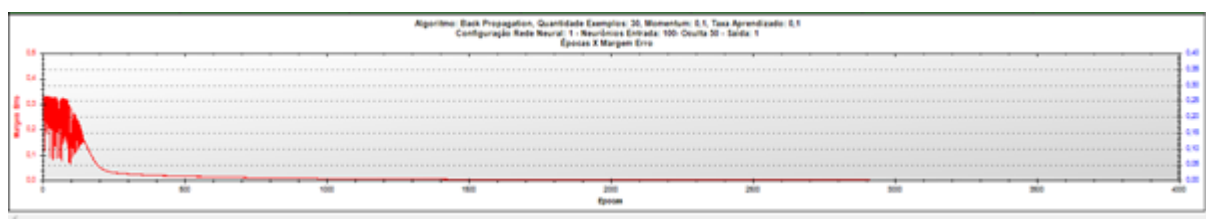
Na configuração da regra do valor médio para determinar o número de neurônios presentes na camada oculta, foram analisados como os treinamentos se comportaram com esta configuração.

#### 4.2.1. UTILIZANDO 10 IMAGENS

Sendo utilizado 10 exemplos de cada vogal (A, E, I, O, U), nessa configuração da RNA, com base na tabela 2 pode-se observar que as melhores RNAs alcançaram uma taxa de acerto de 46% utilizando estas configurações e foi necessário 3200 épocas para realizar o treinamento, conforme é demonstrado na figura 7.

**Tabela 2. Tabela de resultados dos treinamentos com a configuração 1 utilizando 10 exemplos.**

Informações Treinamento					
Treinamento	Épocas				
1106	3200				
Acertos Obtidos Treinamento					
A	E	I	O	U	Total
10/10	10/10	10/10	10/10	10/10	50/50 100%
Acertos Obtidos Validação					
A	E	I	O	U	Total
05/10	04/10	05/10	03/10	06/10	23/50 46%



**Figura 7. Número de épocas necessárias no treinamento de 10 imagens utilizando a configuração 1.**

Fonte: Elaborada pelo autor

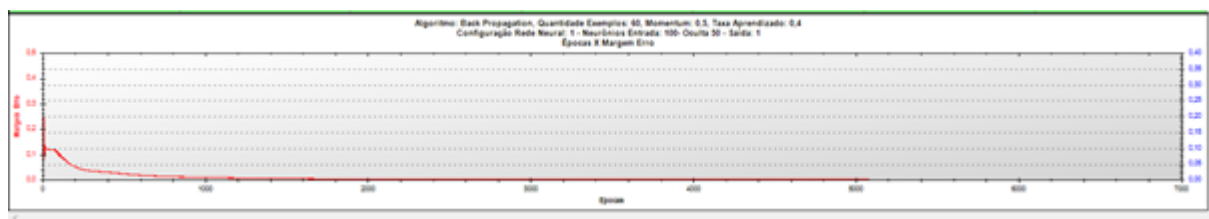
#### 4.2.2. UTILIZANDO 20 IMAGENS

Sendo utilizado 20 exemplos de cada vogal (A, E, I, O, U), nessa configuração da RNA, com base na tabela 3 pode-se observar que as melhores RNAs alcançaram uma taxa de acerto de 39% utilizando estas configurações e foi necessárias 5692 épocas para realizar o treinamento, conforme é demonstrado na figura 8.

**Tabela 3. Tabela de resultados dos treinamentos com a configuração 1 utilizando 20 exemplos.**

001

Informações Treinamento					
Treinamento	Épocas				
1264	5962				
Acertos Obtidos Treinamento					
A	E	I	O	U	Total
20/20	20/20	20/20	20/20	20/20	100/100 100%
Acertos Obtidos Validação					
A	E	I	O	U	Total
06/20	03/20	12/20	11/20	07/20	39/100 39%



**Figura 8. Número de épocas necessárias no treinamento de 20 imagens utilizando a configuração 1.**

Fonte: Elaborada pelo autor

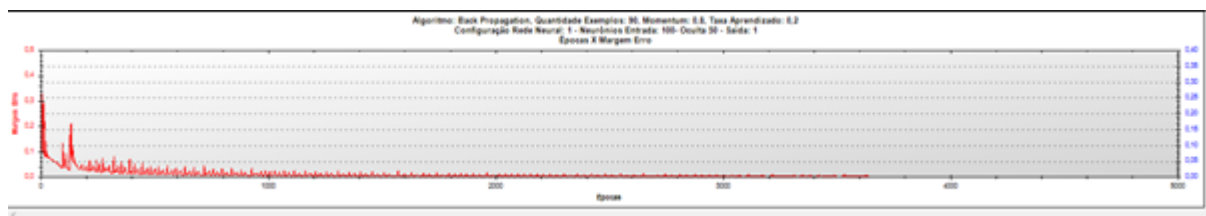
### 4.2.3. UTILIZANDO 30 IMAGENS

Sendo utilizado 30 exemplos de cada vogal (A, E, I, O, U), nessa configuração da RNA, com base na tabela 4 pode-se observar que as melhores RNAs alcançaram uma taxa de acerto de 92% utilizando estas configurações e foi necessárias 3646 épocas para realizar o treinamento, conforme é demonstrado na figura 9.

**Tabela 4. Tabela de resultados dos treinamentos com a configuração 1 utilizando 30 exemplos.**

001

Informações Treinamento					
Treinamento	Épocas				
1366	3646				
Acertos Obtidos Treinamento					
A	E	I	O	U	Total
30/30	30/30	29/30	30/30	29/30	148/150 99%
Acertos Obtidos Validação					
A	E	I	O	U	Total
20/30	20/30	20/30	20/30	20/30	100/150 66%



**Figura 9. - Número de épocas necessárias no treinamento de 30 imagens utilizando a configuração 1.**

Fonte: Elaborada pelo autor

### 4.3. CONFIGURAÇÃO 2 - UTILIZANDO A REGRA DA RAIZ QUADRADA

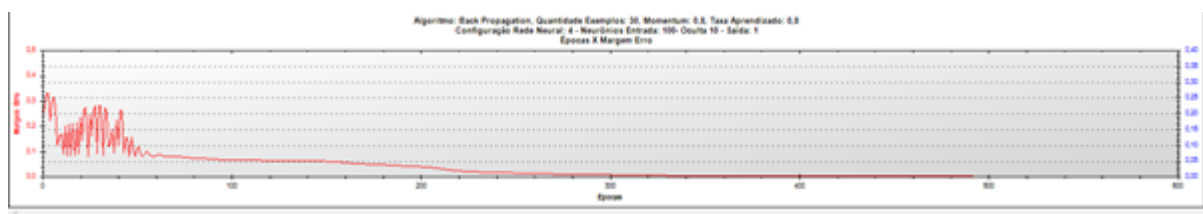
Na configuração da regra da regra da raiz quadrada para determinar o número de neurônios presentes na camada oculta, foram analisados como os treinamentos se comportaram com esta configuração.

#### 4.3.1. UTILIZANDO 10 IMAGENS

Sendo utilizado 10 exemplos de cada vogal (A, E, I, O, U), nessa configuração da RNA, com base na tabela 5 pode-se observar que as melhores RNAs alcançaram uma taxa de acerto de 70% utilizando estas configurações e foi necessárias 1553 épocas para realizar o treinamento, conforme é demonstrado na figura 10.

**Tabela 5. Tabela de resultados dos treinamentos com a configuração 2 utilizando 10 exemplos.**

Informações Treinamento					
Treinamento	Épocas				
1553	515				
Acertos Obtidos Treinamento					
A	E	I	O	U	Total
10/10	09/10	10/10	10/10	10/10	49/50 99%
Acertos Obtidos Validação					
A	E	I	O	U	Total
06/10	06/10	11/10	06/10	06/10	35/50 70%



**Figura 10. Número de épocas necessárias no treinamento de 10 imagens utilizando a configuração 2.**

Fonte: Elaborada pelo autor

#### 4.3.2. UTILIZANDO 20 IMAGENS

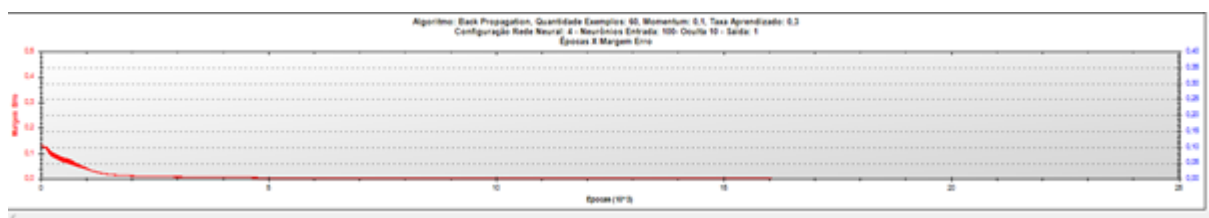
Sendo utilizado 20 exemplos de cada vogal (A, E, I, O, U), nessa configuração da RNA, com base na tabela 6 as melhores RNAs alcançaram uma taxa de acerto de 44% utilizando estas configurações e foi necessárias 18518 épocas para realizar o treinamento, conforme é demonstrado na figura 11.



**Tabela 6. Tabela de resultados dos treinamentos com a configuração 2 utilizando 20 exemplos.**

001

Informações Treinamento					
Treinamento	Épocas				
1612	18518				
Acertos Obtidos Treinamento					
A	E	I	O	U	Total
20/20	20/20	20/20	20/20	20/20	100/100 100%
Acertos Obtidos Validação					
A	E	I	O	U	Total
10/20	02/20	11/20	11/20	10/20	44/100 44%



**Figura 11. Número de épocas necessárias no treinamento de 20 imagens utilizando a configuração 2.**

Fonte: Elaborada pelo autor

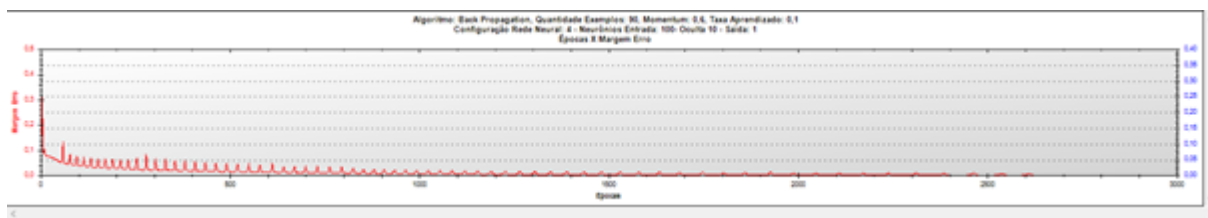
### 4.3.3. UTILIZANDO 30 IMAGENS

Sendo utilizado 30 exemplos de cada vogal (A, E, I, O, U), nessa configuração da RNA, com base na tabela 7 as melhores RNAs alcançaram uma taxa de acerto de 33% utilizando estas configurações e foi necessárias 2629 épocas para realizar o treinamento, conforme é demonstrado na figura 12.

**Tabela 7. Tabela de resultados dos treinamentos com a configuração 2 utilizando 30 exemplos.**

001

Informações Treinamento					
Treinamento	Épocas				
1716	2629				
Acertos Obtidos Treinamento					
A	E	I	O	U	Total
30/30	30/30	30/30	30/30	29/30	146/150 99%
Acertos Obtidos Validação					
A	E	I	O	U	Total
12/30	09/30	07/30	09/30	13/30	50/150 33%



**Figura 12. - Número de épocas necessárias no treinamento de 30 imagens utilizando a configuração 2.**

Fonte: Elaborada pelo autor

## 5. CONCLUSÃO

Com a execução deste trabalho foi possível perceber que a tarefa de desenvolver uma RNA mesmo que seja para algo mais simplório não é algo trivial.

Analisando os resultados, pode-se perceber que para a primeira configuração da RNA que era a configuração utilizando a regra do valor da média, observou-se que a taxa de acerto da RNA foi aumentando gradativamente com a quantidade de exemplos inseridos no treinamento da RNA. Com isso pode-se perceber que quanto mais exemplos inseridos no momento de treinamento da RNA maior será sua taxa de acerto. Agora para a segunda configuração que era a configuração utilizando a regra da raiz quadrado, observou-se que a taxa de acerto da RNA foi o inverso da primeira configuração, pois a taxa de acerto da RNA foi diminuindo gradativamente conforme a quantidade de exemplos foi aumentando.

Entretanto para as respectivas configurações 1 (regra do valor médio) e 2 (regra da raiz quadrada) com as quantidades de exemplos 10, 20 e 30, o melhor treinamento resultante da RNA, com a maior taxa de acerto se deu com a configuração 2 com apenas 10 exemplos, a taxa de acerto no treinamento utilizando esta configuração, ficou com 49 acertos em 50 (99%) e sua taxa de acerto na validação ficou com 35 acerto em 50 (70%).

Possíveis maneiras para se otimizar os resultados da RNA nas atuais condições, poderiam ser alterados os dados de treinamento da RNA, utilizando imagens com um maior grau de características similares as vogais e com uma maior quantidade de exemplos tanto para o treinamento da RNA, quanto para a validação da mesma, também realizando mais treinamentos da RNA utilizando mais configurações, assim buscando obter uma maior taxa de acerto da RNA.

## **6. TRABALHOS FUTUROS**

Este trabalho pode ser aperfeiçoado de diversas formas, como, adicionando mais letras a serem reconhecidas pela rede neural artificial, utilizar novos algoritmos de treinamentos com mais métodos de configurações e novas formas de validações. Pode-se também aumentar o número de formas disponíveis para realizar o treinamento e validação dos dados, também utilizando outros elementos na predição, como formas geométricas e até mesmo o reconhecimento de palavras.

## Referências

- BRAGA, A. P.; CARVALHO, A. C. P. L. F.; LUDEMIR, T. B. *Redes Neurais Artificiais: Teoria e aplicações*. [S.l.]: Rio de Janeiro: LTC, 2000.
- CRUZ, A.; NEHME, D. *Reconhecimento de caracteres usando redes neurais*. [S.l.], 1993.
- DIMAURO, G. et al. Handwriting recognition: State of the art and future trends. In: *Advances in Document Image Analysis*. [S.l.]: Springer, 1997. p. 1–18.
- FAUSSET, L. *FUNDAMENTAL OF NEURAL NETWORKS*. [S.l.]: New Jersey, Prentice Hall, 1994.
- HEATON, J. Encog: library of interchangeable machine learning models for java and c#. *Journal of Machine Learning Research*, v. 16, p. 1243–1247, 2015.
- HINTON, G. E. Connectionist learning procedures. *Artificial intelligence*, Elsevier, v. 40, 1989.
- KOSKO, B. Bidirectional associative memories. *IEEE Transactions on Systems, man, and Cybernetics*, IEEE, v. 18, n. 1, p. 137–144, 1988.
- LIPPMANN, R. An introduction to computing with neural nets. *IEEE Assp magazine*, IEEE, v. 4, n. 2, p. 4–22, 1987.
- OLIVEIRA, F. R. *ESTUDO COMPARATIVO ENTRE OS ALGORITMOS PARA TREINAMENTO DE REDES NEURAIS ARTIFICIAIS: BACKPROPAGATION E RESILIENT PROPAGATION*. [S.l.], 2013.
- RODRIGO, M. C. *PROTOTIPO DE UM SISTEMA DE RECONHECIMENTO DE CARACTERES BASEADO EM REDES NEURAIS*. [S.l.], 2000.
- RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. *Learning internal representations by error propagation*. [S.l.], 1985.
- SILVA, B. M. da; VANDERLINDE, M. *INTELIGÊNCIA ARTIFICIAL, APRENDIZADO DE MÁQUINA*. [S.l.], 1993.
- TAFNER, M.; XEREZ, M.; R., F. *REDES NEURAIS ARTIFICIAIS*. [S.l.]: Blumenau, Furb, 1995.
- TURBAN, E. *Decision support and expert systems: management support systems*. [S.l.]: Prentice Hall PTR, 1990.