

Reconhecimento Ótico de Caracteres Usando Rede Neural *Multilayer Perceptron*

Sávio S. Dias¹, Italo G. Santana¹, Presley M. N. Costa¹

¹Curso de Ciência da Computação – Fundação Universidade Federal do Tocantins
(UFT)

Endereço: Av. NS 15, ALC NO 14, 109 Norte

{diasssavio, italo.gomes.santana}@gmail.com, presley_mc@hotmail.com

Abstract. *This paper describes a neural network based solution that solves the determinants, which are related to optical character recognition numeric systems found in OCR - Optical Character Recognition - much studied, but solved through other applicable solutions to the problem.*

Resumo. *Este artigo descreve uma solução baseada em rede neural, que resolva os determinantes, que são relativos ao reconhecimento ótico de caracteres numéricos, encontrados nos sistemas OCR – Optical Character Recognition – muito estudados, porém resolvidos através de outras soluções aplicáveis ao problema.*

1. Introdução

Por séculos o homem busca entender os comportamentos inteligentes observados em animais ou seres racionais como o próprio ser humano. De maneira empírica notou-se que o comportamento inteligente na sua natureza é na verdade uma técnica de aprendizado desenvolvida ao longo da evolução genética dos seres dotados de tal. E para artificializar esta técnica foram desenvolvidos diversos estudos que investigam a simulação de tal comportamento.

OCR – *Optical Character Recognition* – tecnologia já difundida entre aplicações comerciais tornou-se desafio no início da década de cinquenta quando surgiram os primeiros algoritmos para softwares OCR's. Essa tecnologia destacava-se por permitir o reconhecimento de caracteres a partir de mapas de bits ou arquivos de imagem. Analogamente ao comportamento inteligente essa tecnologia foi desenvolvida também utilizando redes neurais propondo métricas robustas da simulação de inteligência ainda que de forma deficiente se comparado ao cérebro humano. E o problema proposto neste artigo diz respeito à solução de um sistema OCR utilizando redes neurais.

Um dos estudos desenvolvidos sobre OCR descreve uma técnica baseada em redes neurais chamada Rede Neural *Multilayer Perceptron*. Esta foi inspirada na estrutura real das redes neurais, que são formadas por redes de neurônios interligados e semelhantemente ao observado no cérebro que se baseia no aprendizado. E este é realizado por meio de tentativas e erros, onde o neurônio irá adaptar seus processamentos de acordo com os erros obtidos.

O modelo do neurônio artificial foi projetado e adaptado para descrever a simplificação de um neurônio biológico. Cada neurônio possui n entradas e uma saída.

E para cada entrada é associada um peso, que será balanceado de acordo com o erro de saída, pois quando a saída não obtém o resultado esperado percebe-se a variação de um erro.

2. Revisão de literatura

Na Rede Neural *Multilayer Perceptron* constituída apenas de três camadas - entrada, oculta e saída - a camada de entrada é referente aos valores de entrada que os neurônios da camada posterior irão receber, chamada de camada oculta. Estes valores são informações de estímulo para o processamento neural, este responsável por gerar uma saída para cada neurônio da camada de saída.

Os valores das saídas precisam ser tratados para que seja possível determinar uma saída final processada ou quando se estiver na fase de treinamento será calculado o erro obtido, as saídas dos neurônios tornam-se entradas em cada neurônio da camada posterior, a camada de saída, que gerará valores de saída e um erro de saída da rede. Este será utilizado para definir a precisão do processamento ou para iniciar uma etapa chamada *backward*, que realiza o balanceamento dos pesos das ligações contidas na rede propagando a influência dos erros que a saída final sofreu.

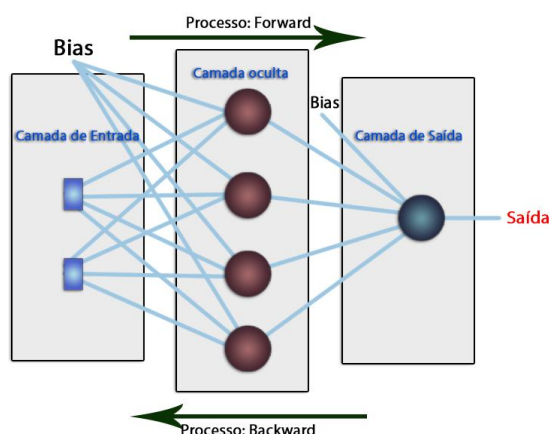


Figura 1. Estrutura funcional da Rede Neural *Multilayer Perceptron*.

De acordo com o representado na Figura1 a etapa *backward* inicia-se na camada de saída e propaga o erro da saída da rede até a camada de entrada, que possui os pesos das ligações de entrada. O processo inverso àquele mencionado é o *forward*, que se inicia na camada de entrada passando pela camada oculta e finalizando com o valor de saída obtido na camada de saída. Essas etapas constituem o método de aprendizado chamado *backpropagation* baseado no aprendizado supervisionado por correção de erros, adotado neste artigo.

Cada neurônio da rede precisa tratar as entradas, que recebe, e processar a saída. Para tanto existem funções matemáticas para balanceamento dos pesos na fase de treinamento e para calcular a saída baseada nas entradas submetidas a seus pesos. A representação funcional do neurônio é básica e consiste apenas na captura dos valores de entrada e o *bias* - utilizado para variar o grau de liberdade, porém permitindo a adaptação ao conhecimento submetido à rede neural - acionando a função aditiva e com

o valor obtido aciona a função de ativação, para gerar a saída daquele neurônio, são elas representadas esquematicamente no modelo funcional do neurônio na Figura 2.

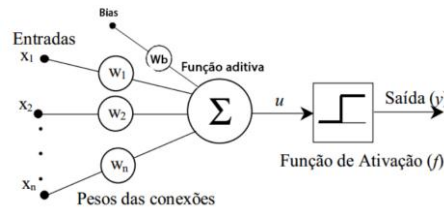


Figura 2. Estrutura funcional de um neurônio

Na fase de treinamento, é crucial especificar como se dá o balanceamento dos pesos nas entradas, que ocorre na etapa *backward*, e para tanto se faz necessário definir as funções que calculam o erro no valor de saída. E existem dois momentos em que as formas são diferentes. No erro da saída da rede neural deve ser observado que foi influenciado pelos valores de saídas da camada oculta, por tanto os balanceamentos dos pesos anteriores a esta camada devem ser calculados de forma particular de acordo com as funções representadas na Figura 3.

$e_j = s - y_j$	<i>s</i> : Saída esperada
$\delta_j = (1 - y_j^2) * e_j$	e_j : Erro de saída da camada de saída
$\delta_{b_j} = (1 - y_j^2) * \sum \delta_i * w_i$	δ_j : Erro Retropropagado da camada de saída
$w_j = w_j + \alpha y_j x_j \delta_j$	w_j : Balanceamento da camada de saída
$w_j = w_j + \alpha y_j x_j \delta_{b_j}$	δ_{b_j} : Erro Retropropagado da camada oculta
	" δ_i " e " w_i " pertencem a camada de saída
	w_j : Balanceamento da camada oculta
	α : Taxa de aprendizado
	x_j : Valores de entrada

Figura 3. Funções matemáticas utilizadas no calculo do erro da camada oculta

3. Descrição do problema

O problema propõe a utilização de uma Rede Neural *Perceptron* de Múltiplas Camadas que consiga reconhecer padrões de números naturais que estão entre 0 e 9, inclusive. Estes padrões são definidos em matrizes 7x5 que possuem trinta e cinco células com valores entre -1 e 1 em cada célula.

Assim, é necessário que se o treinamento da rede neural com todos os padrões possíveis para que ela se comporte da maneira mais próxima possível da resposta esperada. Então é realizada várias iterações em cima de um conjunto de entradas e saídas esperadas para todos os padrões que se desejam testar, a fim de refinar cada vez mais a resposta da rede neural.

Dessa forma, a camada de entrada da rede neural corresponde a um conjunto de 35 entradas simultâneas, que representa a matriz supracitada, mais o BIAS. Cada entrada corresponde a uma célula da matriz, e que tem seu valor definido em 1, para preenchido, e -1, para vazio, sendo o BIAS com valor fixo de 1.

Por sua vez, a camada de saída possui um neurônio para cada resposta possível. No entanto, o resultado final da rede neural é dado pelo maior valor dentro do conjunto de valores de todos os neurônios desta camada. Estes valores pertencem ao intervalo real $[-1,1]$ e que o padrão treinado ao neurônio que representa este valor é dado como resposta ao usuário.

4. Solução adotada

De acordo com análises empíricas realizadas durante os estudos do comportamento da rede neural, a configuração que demonstrou melhores resultados foi a que obteve os valores que foram utilizados na taxa de aprendizado, número de iterações e número de neurônios na camada oculta que foram 0,1; 20.000; e 9, respectivamente. No entanto, os pesos de cada entrada foram definidos aleatoriamente, respeitando sempre o intervalo real $[-1, 1]$. É importante deixar claro que nesta implementação foi suficiente usar apenas uma camada oculta.

Na figura 1 temos o exemplo do reconhecimento de uma entrada de uma variação do número 4 feito pela aplicação OCR, os resultados obtiveram sucesso.

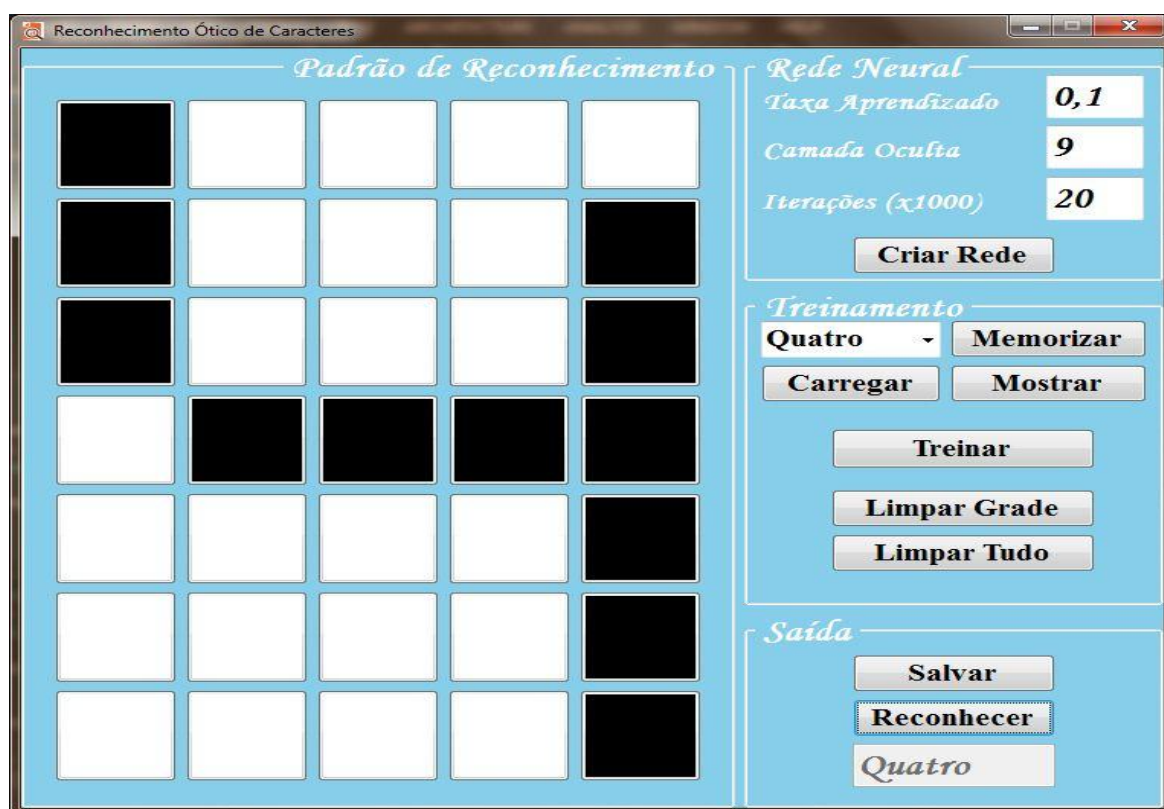


Figura 4. Reconhecimento de uma variação do número 4 treinado

5. Conclusão

Por fim, a Rede Neural *Multilayer Perceptron* utilizada no OCR se mostrou bastante eficiente no que diz respeito às saídas esperadas nos testes feitos em padrões não treinados. Porém, vale ressaltar que em alguns casos, como o exemplo do número zero, com alterações sutis, houve resultados a favor do número oito, isso se deve ao fato,

principalmente, por serem bastante semelhantes na configuração de valores contidos nas células.

No entanto, a grande dificuldade de possuir as melhores configurações de pesos está em achar os valores que melhor se adaptam entre si dentro da rede neural.

Em todos os testes realizados, todos os pesos foram inicializados com valores aleatórios, o que significa o fato de gerar resultados levemente diferentes a cada nova execução.

Referências Bibliográficas

- UNICAMP. (1999) “Redes Neurais”,
ftp://ftp.dca.fee.unicamp.br/pub/docs/vonzuben/theses/lnunes_mest/cap2.pdf
Fevereiro.
- Caraciolo, Marciel P. (2009) “[Artigo].Introduzindo a rede neural Multi-Layer-Perceptron (MLP)”, Fevereiro.
- Haykin, Simon (2001) “Redes neurais: princípios e prática”, Bookman, Editado por Paulo Martins Engel.