

OCR utilizando Redes Neurais e otimização por enxame de partículas (PSO)

Célia Cristina Bojarczuk Fioravanti¹, Wellton Costa de Oliveira¹

¹Universidade Tecnológica Federal do Paraná (UTFPR)
Programa de Pós-Graduação de Engenharia Elétrica e Informática Industrial
(CPGEI) – Curitiba/pr – Brazil

{celiacristina, wcoliveira}@utfpr.edu.br

Abstract. *Os sistemas de reconhecimento ótico, de caracteres OCR (Optical Character Recognition) são sistemas desenvolvidos para, de uma certa forma, reproduzir a capacidade humana de ler textos. O Objetivo deste trabalho é resolver o problema de OCR utilizando duas técnicas muito importantes: Redes Neurais Artificiais multicamadas com o algoritmo de treinamento backpropagation e utilizando otimização por enxame de partículas (PSO). Trinta e seis imagens de dimensões 25x25 foram utilizadas para treinar a rede neural e trinta e duas imagens para testar. No PSO, foram utilizadas imagens de tamanhos aproximados de 170x170,. A rede neural conseguiu acertar 28 letras de 36, errando 4 letras. O PSO se mostrou um método de otimização flexível, onde se pode alterar facilmente a função de fitness a ser utilizada.*

Resumo. *Os sistemas de reconhecimento ótico, de caracteres OCR (Optical Character Recognition) são sistemas desenvolvidos para, de uma certa forma, reproduzir a capacidade humana de ler textos. O Objetivo deste trabalho é resolver o problema de OCR utilizando duas técnicas muito importantes: Redes Neurais Artificiais multicamadas com o algoritmo de treinamento backpropagation e utilizando otimização por enxame de partículas (PSO). Trinta e seis imagens de dimensões 25x25 foram utilizadas para treinar a rede neural e trinta e duas imagens para testar. No PSO, foram utilizadas imagens de tamanhos aproximados de 170x170,. A rede neural conseguiu acertar 28 letras de 36, errando 4 letras. O PSO se mostrou um método de otimização flexível, onde se pode alterar facilmente a função de fitness a ser utilizada.*

1. Introdução

Um sistema de reconhecimento óptico de caracteres (OCR) pode ser apresentado para reproduzir a capacidade humana de ler textos e é composto por uma parte de extração de características e uma parte da classificação [Sakano et al. 1996].

Utilizando-se técnicas de reconhecimento de padrões (Rede neural e otimização por enxame de partículas - PSO), o computador consegue identificar qual letra está naquela imagem específica.

1.1. Objetivos

O objetivo geral deste trabalho é realizar o OCR utilizando redes neurais e PSO.

2. Fundamentação Teórica

2.1. Sistemas de Reconhecimento Ótico - OCR

Os sistemas de reconhecimento ótico, de caracteres OCR (Optical Character Recognition) são sistemas desenvolvidos para, de uma certa forma, reproduzir a capacidade humana de ler textos. Este capítulo abordará as diferentes tópicos relacionados com a assunto, apresentando uma classificação dos sistemas de reconhecimento de caracteres em geral (não somente os óticos) e descrevendo as principais características e algoritmos empregados em tais sistemas[Osorio 1991].

Em geral, os sistemas de OCR possuem implementadas as seguintes funções: aquisição da imagem do texto, tratamento da imagem, localização e separação dos caracteres, pre-processamento dos padrões, extração de atributos, reconhecimento/classificação dos padrões e pós-processamento. A Figura 1 mostra um exemplo de extração de atributos de uma imagem.

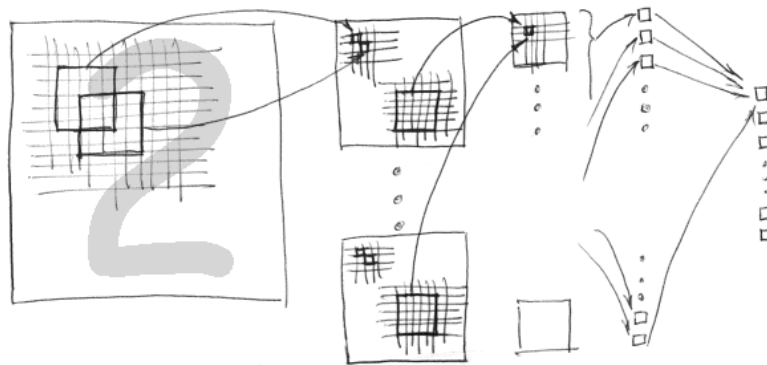


Figura 1. extração de dados (pixels) de uma imagem

uma vez extraídas as características das imagens, é necessária a realização do reconhecimento/classificação dessas imagens para saber qual letra está naquela imagem. para isso, utiliza-se técnicas de reconhecimento de padrões. Uma delas é a Rede neural e outra é a técnica de otimização por enxame de partículas (PSO), que serão vistos nas próximas seções.

2.2. Redes Neurais - Backpropagation

O cérebro humano possui mais de dez bilhões de neurônios, cada qual conectado, em média, a milhares de outros neurônios. Estas conexões são conhecidas por sinapses e o cérebro humano possui cerca de 60 trilhões destas conexões. Neurônios são, na verdade, elementos de processamento muito simples. Cada neurônio é composto por uma soma, que é o corpo do neurônio, um axônio e vários dendritos (Figura 2).

O neurônio recebe entradas de outros neurônios, junto aos seus dendritos e quando este sinal de entrada excede um limiar, o neurônio é "ativado" - na verdade, ocorre uma reação química que resulta em um pulso elétrico, conhecido como potencial de ação, que será enviado ao axônio (a saída do neurônio), em direção às sinapses que conectam um neurônio aos dendritos de outros neurônios.



Figura 2. Um neurônio no cérebro humano

Embora cada neurônio isoladamente seja muito simples, esta enorme e complexa rede de neurônios é capaz de processar informações de extraordinária complexidade, a uma grande taxa. Em termos de complexidade, o cérebro humano excede, de longe, qualquer dispositivo criado pelo homem, ou até mesmo qualquer outro objeto ou estrutura que ocorra naturalmente no universo, pelo que conhecemos atualmente.

O cérebro humano tem uma propriedade chamada plasticidade, o que significa que os neurônios podem mudar a natureza e o número de conexões com outros neurônios em resposta a eventos que ocorram. Deste modo, o cérebro é capaz de aprender. O cérebro usa uma forma de atribuição de crédito para reforçar conexões entre neurônios que levem a soluções corretas para problemas e para enfraquecer conexões que levam a soluções incorretas. A intensidade de uma conexão, ou sinapse, determina qual será sua influência nos neurônios desta conexão e, assim, se uma conexão for enfraquecida, seu papel será atenuado nas computações subsequentes.

Redes de neurônios artificiais são modeladas em relação ao cérebro humano e são compostas por vários neurônios artificiais. Neurônios em redes neurais artificiais tendem a ter menos conexões do que neurônios biológicos e as redes neuronais (atualmente) são todas significativamente menores, em termos de números de neurônios, do que as do cérebro humano. Os neurônios são conhecidos como neurônios de McCulloch e Pitts, que foram os biólogos criadores das redes neurais artificiais (criado em 1943).

Cada neurônio (ou **nó**) em uma rede neural recebe uma série de entradas. Uma função chamada de função de ativação é aplicada a esses valores de entrada, o que resulta no nível de ativação do neurônio, que é o valor de saída do neurônio onde um eixo (x) representa o valor de entrada para o neurônio e um eixo (y) representa a saída ou o nível de ativação do neurônio. (Figura 3).

Uma função das mais comumente usadas é a função degrau ou função de limiar. Ao usar esta função, as entradas para o neurônio são somadas (tendo cada uma sido multiplicada por um peso w), t . Se a soma for maior do que o limiar, o neurônio ativar-se-á e terá um nível de ativação de +1. Caso contrário, ficará inativo e terá zero como nível de ativação (em algumas redes, quando a soma não excede o limiar, considera-se que o nível de ativação seja -1 em vez de 0). Então, o comportamento do neurônio pode ser expresso da seguinte forma

$$X = \sum_{i=1}^n w_{ij}x_i$$

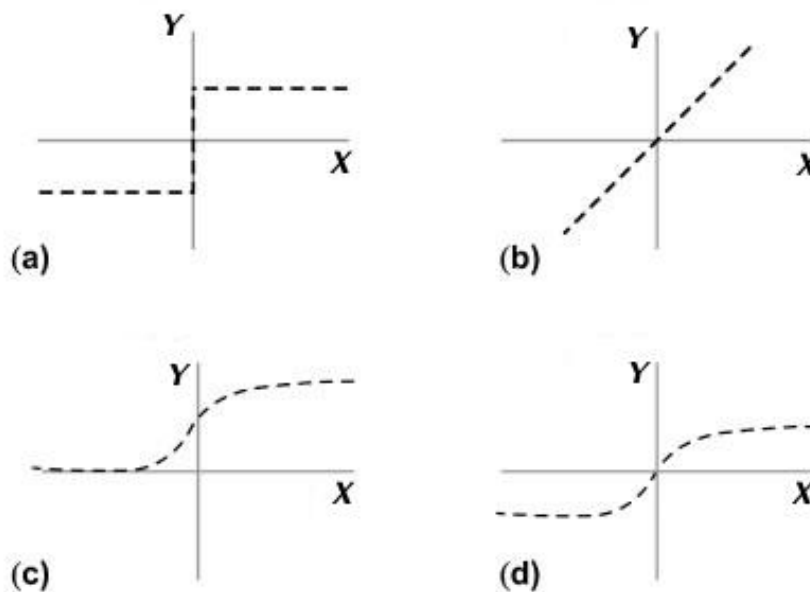


Figura 3. Quatro funções de ativação. (a) Função degrau; (b) função linear; (c) funções sigmoide; (d) função tangente hiperbólica.

X é a soma ponderada das n entradas do neurônio, de x_1 a x_n , onde cada entrada x_i é multiplicada pelo seu peso correspondente w_i . Por exemplo, vamos considerar um neurônio simples que só tenha duas entradas. Cada uma dessas entradas tem um peso associado a ela, como por exemplo $w_1 = 0,8$ e $w_2 = 0,4$, e as entradas de neurônios sendo $x_1 = 0,7$ e $x_2 = 0,9$, então, o peso somado dessas entradas é $(0,8 * 0,7) + (0,4 * 0,9) = 0,92$. O nível de ativação Y é definido para este neurônio como:

$$Y = +1 \text{ para } X < t \text{ ou } 0 \text{ para } X \leq t$$

Então, se t for menor ou igual a 0,92, este neurônio será ativado com este conjunto específico de entradas. Caso contrário, ele terá zero como nível de ativação. Uma rede neural consiste em um conjunto de neurônios conectados. As conexões entre neurônios têm pesos associados a elas e cada neurônio transmite sua saída às entradas do neurônio ao qual estiver conectado. Esta saída depende da aplicação da função de ativação às entradas que ele recebe. Deste modo, um sinal de entrada para a rede é processado por toda a rede e uma saída (ou saídas múltiplas) será produzida. Não há processamento central ou mecanismo de controle - toda a rede é envolvida em cada fragmento de computação que ocorre.

Quando uma entrada é dada a uma rede neural, a saída não aparece imediatamente porque leva um período finito de tempo para os sinais passarem de um neurônio ao outro. Em redes neurais artificiais, este tempo é geralmente muito curto, porém, no cérebro humano, as conexões neuronais são surpreendentemente lentas. É apenas a natureza do cérebro, com um enorme grau de paralelismo, que permite a realização de cálculos tão rapidamente.

Para que as redes neurais aprendam, o peso associado a cada conexão (equivalente a uma sinapse no cérebro humano) pode ser alterado em resposta a conjuntos específicos de entradas e de eventos.

O perceptron, que foi proposto primeiramente por Rosenblatt (1958), é um simples neurônio usado para classificar suas entradas em uma de duas categorias. O perceptron pode ter qualquer número de entradas, que às vezes são organizadas em uma grade. Esta grade pode ser usada para representar uma imagem ou um campo de visão, e, então, os perceptrons podem ser usados para realizar classificação de imagem ou tarefas de reconhecimento simples. O perceptron usa uma função degrau que retorna +1 se a soma ponderada das entradas, X , for maior do que o limiar, t , e -1 se X for menor ou igual a t , conforme já foi comentado anteriormente neste trabalho.

Redes Neurais Multicamadas vem resolver problemas do mundo real que não é linearmente separável e, embora perceptrons sejam um modelo interessante para estudar o modo como os neurônios artificiais podem funcionar, é necessária a utilização de multicamadas que é uma técnica com mais capacidade. A primeira camada é a camada de entrada, cada nó nesta camada recebe um sinal de entrada único. Geralmente, os nós nesta camada não são neurônios, mas apenas atuam na passagem dos sinais de entrada para os nós da próxima camada, que é neste caso uma camada oculta ou escondida. Uma rede pode ter uma ou mais camadas ocultas (ou escondidas) que contenham os neurônios que realmente fazem o trabalho. Cada sinal de entrada é passado a cada um dos nós nesta camada e que a saída de cada nó nesta camada é passada a cada nó da camada final, que é a camada de saída. A camada de saída opera a última etapa de processamento e envia sinais de saída. A rede é chamada de rede com alimentação adiante porque os dados são alimentados a partir dos nós de entrada em direção aos nós de saída. Isso contrasta com as redes recorrentes nas quais alguns dados são passados de volta, a partir dos nós de saída na direção dos nós de entrada. (Figura 4).

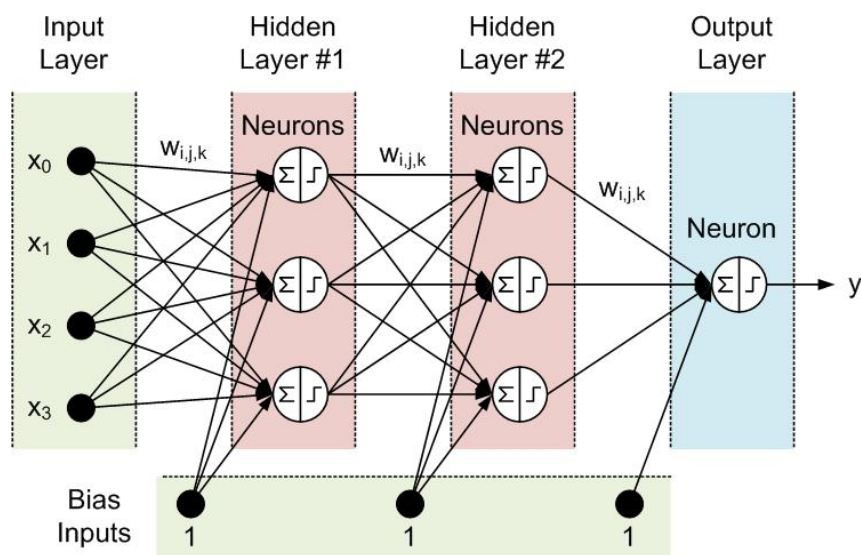


Figura 4. Exemplo de uma rede neural multicamada

Backpropagation (retropropagação) é um método comumente usado em redes neurais artificiais multicamadas onde cada neurônio tem pesos associados em suas entradas e há muito mais pesos a serem ajustados quando um erro é cometido para um fragmento de dados de treinamento. Obviamente, uma questão importante é como atribuir responsabilidade (ou crédito) aos diferentes pesos e esse é o trabalho do backpropagation. Em vez

de usar uma simples função degrau, utilizada por perceptrons únicos, redes multicamadas com backpropagation geralmente usam a função sigmoide.

Assim como o perceptron único, o algoritmo de backpropagation começa atribuindo valores aleatórios aos pesos na rede, que geralmente recebem pequenos valores na faixa entre -0,5 e 0,5. Como alternativa, os pesos podem ser normalmente distribuídos entre $-2,4/n$ e $2,4/n$, onde n é o número de entradas da camada de entrada. Cada iteração do algoritmo envolve, de início, a alimentação de dados por meio da rede, desde as entradas até as saídas. A próxima fase, que dá origem ao nome do algoritmo, envolve a alimentação de erros de volta, das saídas às entradas. Estes valores de erro são alimentados de volta, através da rede, fazendo mudanças nos pesos dos nós ao longo do caminho. O algoritmo segue deste modo, até que as saídas produzidas para os dados de treinamento aproximem-se suficientemente dos valores desejados - em outras palavras, até que os valores de erro sejam suficientemente pequenos. Como a função sigmoide não pode realmente chegar a 0 ou a 1, é comum aceitar valores como 0,9 representando 1 e 0,1 representando 0. Considerando uma rede de três camadas e usar i para representar nós na camada de entrada, j para representar nós na camada oculta e k para representar nós na camada de saída. Então, por exemplo, w_{ij} se refere ao peso de uma conexão entre um nó na camada de entrada e um nó na camada oculta. A função que é usada para gerar o valor de saída de um nó j na rede pode ser vista como:

$$X_j = \sum_{i=1}^n x_i \cdot w_{ij} - \theta_j$$
$$Y_j = \frac{1}{1 + e^{-X_j}}$$

sendo n o número de entradas para o nó j , w_{ij} o peso da conexão entre cada nó i e cada nó j , θ_j é o valor de limiar em uso para o nó j , que é estabelecido em um valor aleatório entre 0 e 1, x_i é o valor de entrada para o nó de entrada i e Y_j é o valor de saída produzido pelo nó j . Uma vez que as entradas tenham sido alimentadas por meio da rede para produzirem saídas, um gradiente de erro é calculado para cada nó k na camada de saída. A iteração do algoritmo de backpropagation é geralmente concluída quando a soma dos quadrados dos erros dos valores de saída, para todos os dados de treinamento em uma época, for menor do que um limiar, tal como 0,001.

2.3. Otimização por Enxame de Partículas - PSO

Omran (2005), em seus estudos investigou a utilização de um método eficiente de otimização, conhecido como PSO (Particle Swarm Optimization), para o campo de reconhecimento de padrões e processamento de imagens. Inicialmente um método de agrupamento baseado no PSO foi proposto, sendo que a aplicação de um algoritmo de agrupamento para o problema de classificação não supervisionada e segmentação de imagens foi investigada. Finalmente, as abordagens baseadas em PSO foram propostas para a quantização da imagem colorida e problemas de separação espectral. Neste estudo também foi avaliada a influência dos parâmetros do PSO sobre o desempenho dos algoritmos propostos.

Ba-Karait e Shamsuddin (2008) utilizaram neste estudo o PSO para reconhecer dígitos escritos à mão sem restrições. Neste estudo cada classe foi codificada sendo o centroide no espaço no espaço de características multidimensionais e o PSO foi utilizado para sondar a posição ideal para cada centroide. Os resultados obtidos revelaram que

PSO forneceu um desempenho promissor e um comportamento estável em reconhecer esses dígitos [Sahel Ba-Karait and Shamsuddin 2008].

Gorai e Ghosh (2009) utilizaram o PSO (Particle Swarm Optimization) neste estudo, uma vez que consideraram o realce da imagem um problema de otimização. O realce da imagem foi feito principalmente para maximizar o conteúdo de informação da imagem melhorada com a função de transformação de intensidade parametrizada, que utilizou a informação local e global da imagem. Um critério objetivo para medir a melhoria da imagem foi utilizado que considerava as informações de entropia e borda da imagem, e tentando conseguir uma melhor imagem foram otimizados os parâmetros utilizando o PSO [Gorai and Ghosh 2009].

Huang e Yin (2010) propuseram a utilização do PSO baseado em rede neural para vídeo OCR, a fim de permitir a recuperação baseada em conteúdo automática na base de dados de vídeo digital. Considerando que o texto é muitas vezes apresentada contra um fundo complexo, o problema desafiador na pesquisa foi a sua detecção e extração. A transformada wavelet foi feita realizada e os coeficientes foram as entradas para a rede neural treinada por PSO. A saída final da rede de regiões de texto foi diferente daquelas regiões não-texto. Os resultados experimentais demonstraram a eficácia do método proposto [Huang and Yin 2010].

Kroetz et al. (2011) propuseram um estudo inicial baseado em otimização por enxame de partículas, em conjunto com registro de imagens para a detecção automática do cordão de solda do tipo PDVD (parede dupla vista dupla – assim denominada pois a radiação precisa atravessar os dois lados da parede da tubulação antes de alcançar o filme radiográfico e em tais situações a fonte fica posicionada a uma distância maior da peça) em radiografias. O PSO foi aplicado para otimizar o registro das imagens modelo, onde as variáveis que determinam as posições do tubo e do cordão são as variáveis otimizadas e compõem os parâmetros das partículas. Segundo Kroetz et al., as imagens utilizadas não apresentavam variações expressivas nas geometrias e posições do tubo e do cordão de solda, portanto, testes adicionais com radiografias que apresentem uma diversidade maior de situações se fazem necessários para testar a generalidade do método. Neste estudo inicial, a utilização de PSO para otimizar o registro de imagens se mostrou eficaz.

Suyama et al. (2013) apresentaram uma metodologia para a detecção da região central de cordões de solda de tubulações de petróleo em imagens radiográficas parede dupla vista dupla. O método foi baseado em três etapas: pré-processamento (para isolar as regiões candidatas), otimização (para definir a elipse que melhor se inscreve em cada região candidata), decisão (escolha da melhor região candidata). Os resultados obtidos mostraram que o algoritmo PSO convergiu de maneira satisfatória para a seleção do polígono que mais se assemelhava à região central do cordão de solda [Huang and Yin 2010].

A otimização por enxame de partículas (Particle Swarm Optimization – PSO) é proposta inicialmente por James Kennedy e Russel Eberhart [Kennedy 2006], como um modelo para otimização de funções contínuas não-lineares. O PSO constitui-se de uma heurística populacional derivada de modelos do comportamento coletivo de animais e insetos na coordenação de seus movimentos em tarefas, tais como: busca e obtenção de alimentos [Heppner and Grenander 1990]. Os modelos iniciais foram simplificados,

perdendo a exigência de manter uma distância mínima de seus vizinhos e transformando a arquitetura de comunicação, a qual era inicialmente baseada em proximidade espacial. A arquitetura foi alterada para usar uma topologia bem definida por um grafo. Assim, o PSO possui, atualmente, mais semelhanças com modelos de influência mútua entre seres humanos em suas formas de pensar e agir [Axelrod 1997].

No PSO, o enxame é constituído por uma quantidade fixa de agentes, chamados de partículas, e estruturado de forma que cada agente seja capaz de comunicar-se com um subconjunto de seus pares, chamados de vizinhos. Cada partícula movimenta-se pelo espaço de soluções com uma determinada velocidade, avaliando a cada iteração a solução correspondente à posição ocupada. Essa velocidade deve sofrer influência das experiências passadas próprias (fator cognitivo) e de seus vizinhos (fator social), sendo essas influências usualmente implementadas como dois atratores, o primeiro localizado na melhor posição já avaliada pela partícula em questão e o segundo localizado na melhor posição visitada pelas partículas adjacentes [Axelrod 1997].

Considerando um espaço de busca N-dimensional e um enxame com S partículas, a posição da i-ésima (i^{th}) partícula do enxame na j-ésima (j^{th}) dimensão pode ser atualizada mediante as equações 3.1 e 3.2 (Kennedy e Eberhart, 1995; Kennedy et al., 2001):

$$v_{ij}(t+1) = w.v_{ij}(t) + c_1.r_1(y_{ij}(t) - x_{ij}(t)) + c_2.r_2(y_{ij}(t) - x_{ij}(t))$$

$$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1)$$

em que w é o peso da inércia que decresce linearmente durante o tempo de execução do algoritmo; r_1 e r_2 são números aleatórios uniformemente distribuídos entre 0 e 1; c_1 e c_2 são constantes que representam o parâmetro cognitivo e o social, respectivamente, com o cognitivo representando a influência do conhecimento individual e, o social, a influência do conhecimento coletivo; y_{ij} é a melhor posição individual da i^{th} partícula na j^{th} dimensão e y_{ij} é a melhor posição global entre todas as partículas na i^{th} dimensão [Axelrod 1997].

Na equação 3.1, a primeira parte é o histórico de movimentação e está relacionada com a estabilidade do enxame de partícula. A segunda parte consiste da parte cognitiva, que representa o conhecimento independente da partícula. A terceira parte é a parte social, que representa a colaboração entre as partículas [Axelrod 1997].

Como resultado dessa dinâmica, ocorre que o enxame, inicialmente disperso por todo o espaço de busca, passa a focar em determinadas regiões que apresentaram soluções mais promissoras com o decorrer da execução do algoritmo. Isso permite que o PSO seja capaz de refinar boas soluções já encontradas, sem perder a capacidade de realizar uma busca mais ampla no espaço. Observa-se que, ao contrário de alguns métodos tradicionais de busca em inteligência artificial, em geral um enxame de partículas avalia uma solução inteira por vez, não fazendo avaliações da contribuição de cada parte para a solução como um todo. Nota-se, também, que uma partícula isolada possui uma limitada capacidade de otimização, sendo altamente propensa a ficar presa em ótimos locais [Axelrod 1997].

A despeito de ser uma meta-heurística relativamente recente, a otimização por enxame de partículas conseguiu granjear um relevante espaço [Poli et al. 2007] devido à sua simplicidade e poder de processamento.

Ao contrário de outras técnicas tradicionais de programação matemática, o PSO

não requer informações de gradiente da função objetivo, necessitando apenas de seus valores nos pontos avaliados e de operadores simples.

Sendo sistemas auto-organizados, os paradigmas de inteligência de enxame, que é uma técnica computacional baseada no comportamento coletivo de sistemas auto-organizados de sistemas descentralizados com o objetivo de realizar uma determinada tarefa, levam a processos altamente distribuídos, com baixa dependência de indivíduos isolados, de forma que o PSO é um modelo de otimização relativamente robusto contra a atração exercida por regiões de baixa qualidade sobre alguns agentes. O fato deste mecanismo centrar-se na ação de agentes baseados em regras muito simples, leva a uma facilidade em sua implementação e também a baixas exigências de memória e poder de processamento [Eberhart et al. 1996]. Estas baixas exigências permitem que cada iteração seja executada em pouco tempo, levando a uma rápida convergência, decorrente também da intensa troca de informação entre os agentes.

No PSO, verifica-se que o balanço entre exploração e busca local, crucial para um bom algoritmo de otimização, é realizado através de um consenso entre um indivíduo e seus vizinhos [Kennedy 2006], sendo o tamanho do passo limitado em cada dimensão pela soma do fator de inércia da partícula às suas distâncias aos atratores pessoal e social. Dessa forma, se um conjunto de partículas conectadas obtiver sucesso em uma determinada região, ele se concentrará nesta porção do espaço e permitirá uma fase de maior exploração; caso contrário, se o grupo encontrar-se disperso pelo espaço de busca, a partícula deverá vagar sem uma tendência definida [Kennedy 2006].

3. Metodologia

Para este trabalho foi fornecido uma base de dados com 258 exemplos para o conjunto de treinamento e 145 exemplos para o conjunto de teste, a fim de classificar o nível de conhecimento do usuário. Os dados constantes na base eram cinco atributos de entrada, sendo eles STG (The degree of study time for goal object materials), SCG (The degree of repetition number of user for goal object materials), STR (The degree of study time of user for related objects with goal object), LPR (The exam performance of user for related objects with goal object) e PEG (The exam performance of user for goal objects); e um atributo de saída, sendo UNS (The knowledge level of user).

3.1. OCR Utilizando Redes Neurais

Para este trabalho, foi utilizado uma rede neural multicamada com o algoritmo de back-propagation para os ajustes de pesos quando um erro é cometido para um fragmento de dados de treinamento. Foi utilizado uma biblioteca em C chamada FANN (Fast Artificial Neural Network). Esta biblioteca cria um arquivo ".net" baseado na configuração da rede (número de neurônios de entrada, de saída, escondidos, época, erro esperado) e baseado também no arquivo de treinamento dado ".train". Baseado neste arquivo ".net", o algoritmo de teste consegue retornar um erro médio que a rede teve baseado em outro arquivo de teste ".test". A metodologia consiste em realizar o treinamento de dados extraídos de 36 imagens de dimensões 25x25 (gerando 625 pixels de entrada), sendo 12 imagens de letras A (diferentes), 12 de imagens de letras B (diferentes) e 12 de imagens de letras P (diferentes), como segue a Figura 5. Os dados aqui referem-se aos valores de pixels de cada posição x y da matriz que forma a imagem, onde cada pixel é comparado

com um limiar (100) onde o pixel que for menor que 100 recebe 0.01 e o pixel que for maior que 100 recebe valor 0.99, tornando assim a imagem binarizada para melhor realizar o treinamento da rede neural. Esses Pixels são extraídos das imagens utilizando a biblioteca openCV.

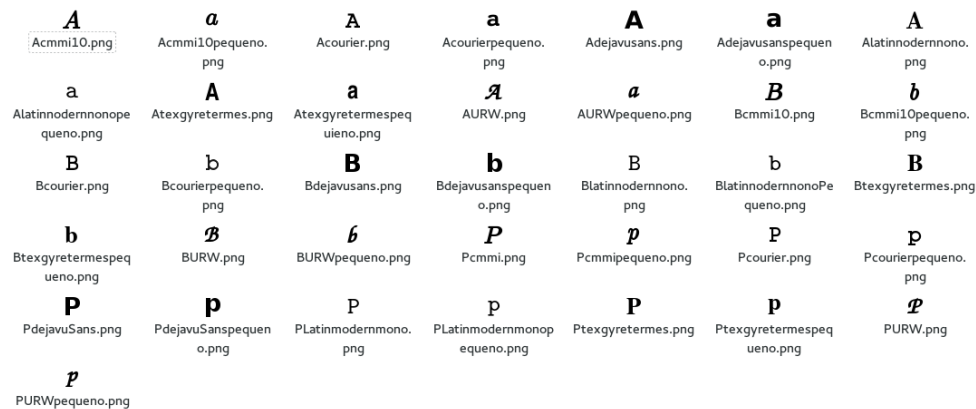


Figura 5. Base de imagens de letras para treinamento da rede neural

Uma base com imagens para teste também foi criada, onde trinta e duas imagens foram criadas, onde 17 imagens são A's, 9 imagens são de B's e 6 imagens são de P's, conforme a Figura 6.

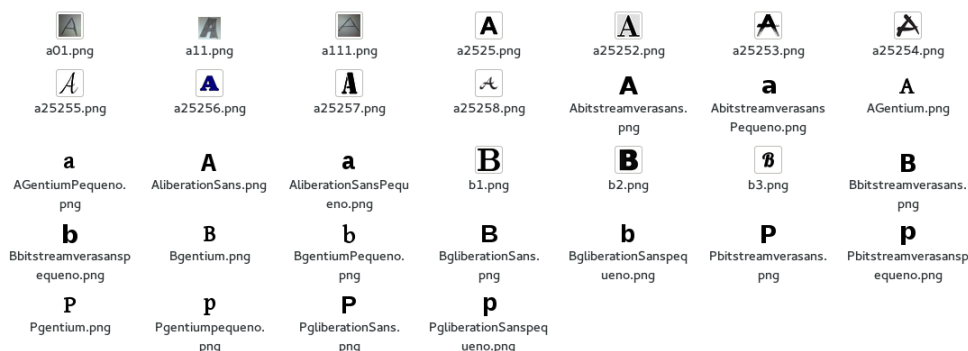


Figura 6. Base de imagens de letras para teste da rede neural

3.2. OCR Utilizando otimização por enxame de partículas (PSO)

Para extrair características da imagem utilizou-se uma parte do processamento de imagens, neste caso, o reconhecimento de padrões para encontrar pontos na imagem que identificassem uma possível letra [?]. Em conjunto com alguns trabalhos pesquisados [?], verificou-se que para esta tarefa utilizaram a distância Euclidiana ou a distância Mahalanobis.

No PSO, foi acrescentado à partícula as coordenadas da imagem, bem como valor do pixel em cada coordenada. A função de fitness utilizada no PSO, levando-se em consideração os artigos lidos, foi a distância euclidiana entre o indivíduo na população e a saída a ser reconhecida.

$$fitness = DE = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

O matlab foi utilizado para verificar o funcionamento do PSO inicialmente, e na sequencia foi acrescentado a imagem a ser analisada para detectar um ponto ou uma região que tivesse na imagem uma letra.

4. Resultados e Discussão

Para os resultados e discussão da implementação do OCR utilizando redes neurais artificiais multicamadas com algoritmo de backpropagation de treinamento, com 625 entradas, um neurônio de saída, uma camada escondida de 100 neurônios e 500 mil épocas, chegou ao erro esperado (de 0.00001) na época 53 (Figura 7).

```
Max epochs 500000. Desired error: 0.0001000000.  
Epochs      1. Current error: 0.1604284346. Bit fail 24.  
Epochs     50. Current error: 0.0001545460. Bit fail 0.  
Epochs     53. Current error: 0.0000958545. Bit fail 0.
```

Figura 7. Resultado do treino da rede neural

com esse treinamento, a rede conseguiu acertar 28 letras de 36, errando 4 letras, conforme a Figura 8

```
teste 1. Resultado 0.0423228. Original 0.01. Erro 0.0323228. acertou  
teste 2. Resultado 0.041198. Original 0.01. Erro 0.031198. acertou  
teste 3. Resultado 0.0496456. Original 0.01. Erro 0.0396456. acertou  
teste 4. Resultado 0.0069941. Original 0.01. Erro 0.0030059. acertou  
teste 5. Resultado 0.884546. Original 0.01. Erro 0.874546. errou  
teste 6. Resultado 0.0308322. Original 0.01. Erro 0.0208322. acertou  
teste 7. Resultado 0.0422595. Original 0.01. Erro 0.0322595. acertou  
teste 8. Resultado 0.0446474. Original 0.01. Erro 0.0346474. acertou  
teste 9. Resultado 0.00926509. Original 0.01. Erro 0.000734907. acertou  
teste 10. Resultado 0.0423245. Original 0.01. Erro 0.0323245. acertou  
teste 11. Resultado 0.0091169. Original 0.01. Erro 0.000883102. acertou  
teste 12. Resultado 0.00707649. Original 0.01. Erro 0.00292351. acertou  
teste 13. Resultado 0.0102959. Original 0.01. Erro 0.000295946. acertou  
teste 14. Resultado 0.0069947. Original 0.01. Erro 0.0030053. acertou  
teste 15. Resultado 0.00911336. Original 0.01. Erro 0.000886642. acertou  
teste 16. Resultado 0.00705871. Original 0.01. Erro 0.00294129. acertou  
teste 17. Resultado 0.00902578. Original 0.01. Erro 0.000974224. acertou  
teste 18. Resultado 0.542486. Original 0.5. Erro 0.0424863. acertou  
teste 19. Resultado 0.950498. Original 0.5. Erro 0.450498. errou  
teste 20. Resultado 0.462527. Original 0.5. Erro 0.0374735. acertou  
teste 21. Resultado 0.68693. Original 0.5. Erro 0.18693. acertou  
teste 22. Resultado 0.987979. Original 0.5. Erro 0.487979. errou  
teste 23. Resultado 0.532624. Original 0.5. Erro 0.0326239. acertou  
teste 24. Resultado 0.501774. Original 0.5. Erro 0.00177389. acertou  
teste 25. Resultado 0.473083. Original 0.5. Erro 0.0269166. acertou  
teste 26. Resultado 0.990719. Original 0.5. Erro 0.490719. errou  
teste 27. Resultado 0.991616. Original 0.99. Erro 0.0016157. acertou  
teste 28. Resultado 0.990511. Original 0.99. Erro 0.000510693. acertou  
teste 29. Resultado 0.992134. Original 0.99. Erro 0.0021342. acertou  
teste 30. Resultado 0.99192. Original 0.99. Erro 0.00192034. acertou  
teste 31. Resultado 0.99205. Original 0.99. Erro 0.00205016. acertou  
teste 32. Resultado 0.989786. Original 0.99. Erro 0.000213861. acertou  
Erro médio quadrado: 0.0899772  
acertos:28  
erros:4
```

Figura 8. Resultado dos testes da rede neural

Para os resultados do PSO, foram utilizadas imagens de tamanhos aproximados de 170x170, podendo ter uma variação conforme a letra na imagem. As imagens a serem

analisadas são em níveis de cinza de 0 a 255. Foi utilizado o valor de 50 para as iterações, a inércia teve o valor inicial de 0.5, o fator de correção de 2.0 e o tamanho da população de 169. Com estes valores iniciais, verificou-se que com o PSO foi possível obter regiões iniciais indicando partes da letra a ser identificada.

Um primeiro exemplo analisado foi a letra P com o tamanho de 136x166, tendo-se obtido uma região no lado esquerdo superior da letra. Outro exemplo foi a letra D com o tamanho de 168x145, tendo-se obtido também uma região no lado esquerdo superior da letra. A Figura 9 ilustra estes resultados, observando-se que na imagem a posição inicial é na parte superior esquerda, enquanto que o resultado do PSO é mostrado na posição inferior esquerda.

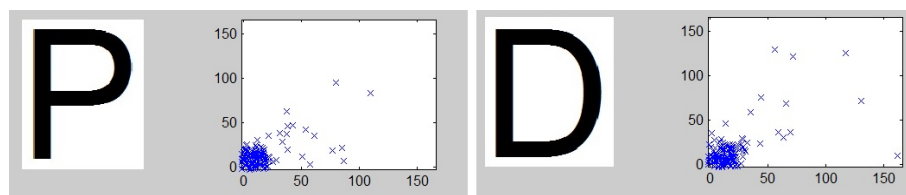


Figura 9. Resultado com a imagem letra P e resultado com imagem letra D

5. Conclusão

Neste trabalho foram realizadas duas soluções diferentes para OCR, uma utilizando Redes Neurais Artificiais multicamadas com o algoritmo de treinamento backpropagation e outra utilizando pso.

Para a rede neural, foi utilizado a biblioteca em C FANN (fast artificial neural network) que possui o algoritmo de backpropagation, onde para uma rede neural com uma camada escondida de 100 neurônios e 500 mil épocas, chegou ao erro esperado (de 0,0001) na época 53 e conseguiu acertar 28 de 32, errando obviamente 4 vezes.

Com este trabalho verificou-se que o PSO é um método de otimização flexível, onde se pode alterar facilmente a função de fitness a ser utilizada. Percebeu-se que alguns pontos não convergiram para a região inicial das letras, mas isto não tira o mérito da convergência do PSO. Com isto, verificou-se que podem ser realizadas modificações na função de fitness, para melhorar o desempenho do PSO.

Referências

- Axelrod, R. (1997). The dissemination of culture a model with local convergence and global polarization. *Journal of conflict resolution*, 41(2):203–226.
- Eberhart, R., Simpson, P., and Dobbins, R. (1996). *Computational intelligence PC tools*. Academic Press Professional, Inc.
- Gorai, A. and Ghosh, A. (2009). Gray-level image enhancement by particle swarm optimization. In *Nature & Biologically Inspired Computing, 2009. NaBIC 2009. World Congress on*, pages 72–77. IEEE.
- Heppner, F. and Grenander, U. (1990). A stochastic nonlinear model for coordinated bird flocks. *AMERICAN ASSOCIATION FOR THE ADVANCEMENT OF SCIENCE, WASHINGTON, DC(USA). 1990*.

- Huang, T. and Yin, L. (2010). Hybrid particle swarm optimized wavelet network for video ocr. In *Information Processing (ISIP), 2010 Third International Symposium on*, pages 91–94. IEEE.
- Kennedy, J. (2006). Handbook of nature-inspired and innovative computing. *Swarm intelligence*, pages 187–219.
- Osorio, F. S. (1991). Um estudo sobre reconhecimento visual de caracteres através de redes neurais.
- Poli, R., Kennedy, J., and Blackwell, T. (2007). Particle swarm optimization. *Swarm intelligence*, 1(1):33–57.
- Sahel Ba-Karait, N. and Shamsuddin, S. M. (2008). Handwritten digits recognition using particle swarm optimization. In *Modeling & Simulation, 2008. AICMS 08. Second Asia International Conference on*, pages 615–619. IEEE.
- Sakano, H., Kida, H., and Mukawa, N. (1996). Seeing the character images that an ocr system sees-analysis by genetic algorithm. In *Pattern Recognition, 1996., Proceedings of the 13th International Conference on*, volume 4, pages 411–416. IEEE.