Integração de Algoritmo Genético com Algoritmo de Classificação Supervisionada (k-NN)

Lucas Souza Santos[1], **Lucas Vinicius Ribeiro**[2]

[1]Departamento de Computação (DACOM) – Universidade Tecnológica Federal do Paraná (UTFPR)
Caixa Postal 271 – 87.301-899 – Campo Mourão – PR – Brazil

{ribeirol, lsantos.2016}@alunos.utfpr.edu.br

Abstract. *This article describes the global optimization experience of a population using a genetic algorithm integrated with a supervised classification algorithm (k-NN). The algorithm seeks, based on an initial population of individuals, to find the features that optimize the hit rate for the digit recognition problem from a training / test set. The Python language was chosen in the implementation of the algorithm in order to facilitate the readability of the problem resolution. The results showed a considerable increase in the hit rate in digit recognition.*

Resumo. *Este artigo descreve a experiência de otimização global de uma população utilizando um algoritmo genético integrado com um algoritmo de classificação supervisionada (k-NN). O algoritmo busca, com base em uma população inicial de individuos, encontrar as características que otimizam a taxa de acerto para o problema de reconhecimento de dígitos a partir de um conjunto de treinamento/teste. A linguagem Python foi escolhida na implementação do algoritmo visando facilitar a legibilidade da resolução do problema. Os resultados evidenciaram um aumento considerável na taxa de acerto no reconhecimento dos dígitos.*

1. Introdução

O presente trabalho aborda a implementação de um algortimo genético integrado ao algoritmo de classificação supervisionada k-NN com intuito de otimizar um modelo de reconhecimento de dígitos a partir de um conjunto pré estabelecido de teste e de treinamento. O artigo está estruturado da seguinte forma: A Seção 2 explica, de maneira geral, o funcionamento do k-NN e aloritmo genético. Em seguida, temos a Seção 3 que trata de apresentar o problema proposto. As Seções 4 e 5 trazem a implementação e os resultados obtidos, respectivamente. E a Seção 6 contém as conclusões.

2. Background

2.1. Algoritmos Genéticos

A associação de técnicas de otimização com ferramentas de Inteligência Artificial proporcionam uma respresentação de situações cada vez mais reais e dinâmicas, pois os algoritmos de busca heurística visam encontrar soluções de alta qualidade em um tempo consideravelmente suportável. Um exemplo de modelo flexível de método heurístico, proposto inicialmente pelo Professor John Holland (1975) é a Programação Genética, quem tem se destacado na solução de diversos problemas de uma maneira simples sem demandar muito conhecimento prévio no assunto.

A ideia de um Algoritmo Genético é seguir as etapas do processo de evolução natural das espécies, incorporando-as a um algoritmo computacional, gerando assim, a partir de uma geração inicial de indivíduos, novos indivíduos com características superiores aos da antiga geração. A busca de soluções em um Algoritmo genético segue um processo evolutivo, onde é feita a seleção dos mais aptos para que seja realizada a transmissão de características dos mesmos para uma nova população através da operação de recombinação e/ou mutação.

2.2. Algoritmo k-NN

Dentre as diversas técnicas empregadas para reconhecer padrões, o algoritmo de classificação baseado no vizinho mais próximo (k-Nearest Neighbor) ou simplesmtente k-NN é amplamente empregado e pertence ao grupo de algoritmos de classificação supervisionada. Seu funcionamento se baseia na descoberta dos k vizinhos mais próximos de uma dada amostra para determinar o rótulo de classificação da mesma. O que determina essa proximidade entre os viznhos é a distância calculada entre os pontos em um espaço euclidiano, onde são utilizadas diversas métricas diferentes, tais como, Distância Euclidiana, City Block, Minkowsky.

2.2.1. Distância Euclidiana

$$\mathrm{d}(\mathrm{x}_i, x_j) = \sqrt[2]{\sum (x_i - x_j)^2}$$

3. Problema

O problema que este trabalho visa resolver, é dado por um conjunto de treinamento e testes que contém características de 10 dígitos numéricos (de 0 a 9). Tanto o conjunto de treinamento quanto o de teste possuem 1000 (mil) linhas, em que cada linha representa uma entrada do problema. Das mil linhas, são 100 linhas para cada um dos dígitos considerados no problema. Além disso, cada linha possui um total de 132 características de um determinado dígito. Sendo assim, o objetivo deste trabalho é utilizar o algoritmo genético integrado ao k-NN para otimizar e melhorar o desempenho do modelo ao reconhcer os dígitos a partir dos conjuntos de treinamento e teste.

4. Implementação

A implementação do Algoritmo Genético e k-NN foi feita em linguagem de programação Python (versão 3). Para resolver o problema, foram adotadas as seguintes métricas: cada população tem um total de 7 indivíduos. A primeira população é gerada aleatoriamente. Para gerar uma nova geração de indivíduos, é mantido o indivíduo de melhor resultado da geração anterior, dois indivíduos gerados pelo cruzamento dos dois melhores da geração anterior, dois indivíduos gerados a partir do cruzamento do melhor com o pior indivíduo da geração anterior e dois indivíduos gerado a partir dos dois piores indivíduos da geração anterior. O operador de mutação é executado em 10 classes aleatórias de indivíduos selecionados randomicamente. Como há um total de 1000 (mil) indivíduos de teste, temos um total de 1% de mutação. Tendo a população formada, ela é usada como entrada para o k-NN, que irá testar as características designadas no indivíduo, comparando as entradas de teste com o conjunto de treinamento, gerando uma porcentagem de acerto.

5. Resultados

Após implementar o algoritmo, executamos o teste com uma população de 7 indivíduos e K = 1 a partir de um conjunto de teste e treinamento. Contudo, tivemos em média 4 minutos de tempo gasto por geração. A primeira geração (criada aleatóriamente) teve como indivíduo de melhor avaliação a partir da classificação k-NN o indivídio 3, com acurácia de 91.3%, e menor avaliação 88.7% pertencente ao indivíduo 7.

No total, foram reproduzidas 43 gerações contando com a inicial, sendo feitas as operações de recombinação e mutação em cada uma das gerações novas. Após a execução e criação das 43 gerações de indivíduos, obtivemos como última geração, uma população de melhor e pior avaliação 94.19% e 86.9% respectivamente. A partir dos resultados obtidos, tivemos um aumento de 2,89% na taxa de acerto.

A Figura 1 apresenta os resultados obtidos nas primeiras gerações do algoritmo genético, enquanto a Figura 2 apresenta os resultados obtidos nas últimas gerações do algoritmo.

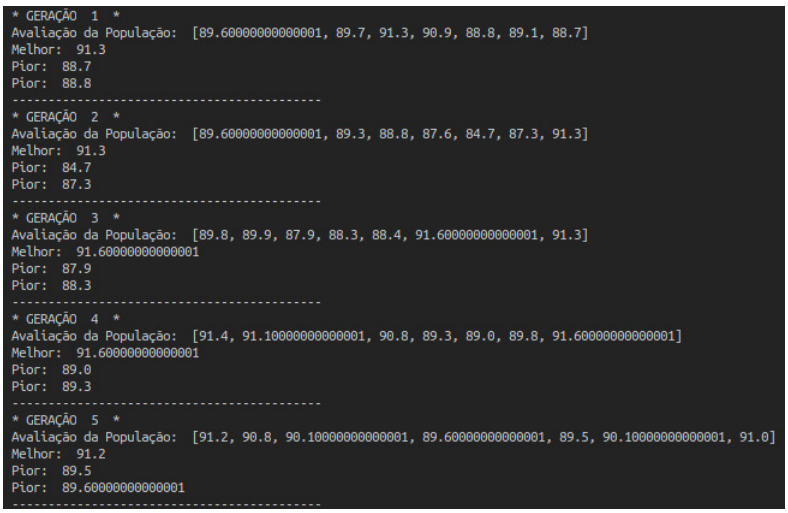

Figura 1. Primeiras gerações do algoritmo genético

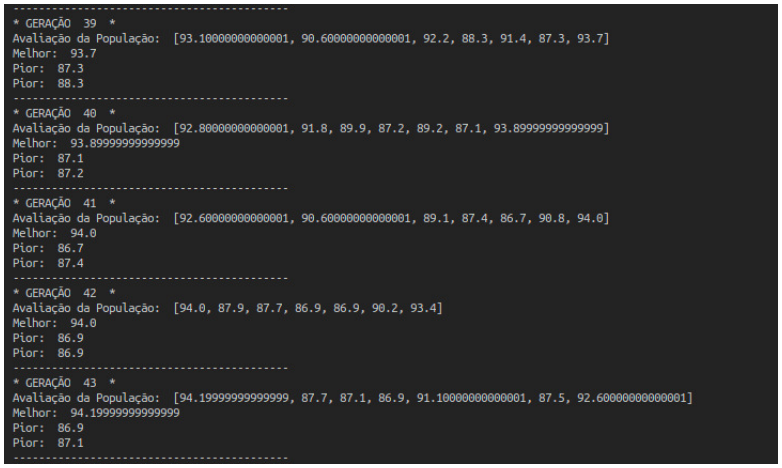

Figura 2. Últimas gerações

6. Conclusões

Com este trabalho foi possível concluir que o uso do Algoritmo Genético pôde melhorar consideravelmente a taxa de acerto para resolução do problema proposto.

Referências

MARQUES, L. *KNN - Nearest Neighbor*. Disponível em: ⟨https://www.wattpad.com/115821129-algoritmos-de-aprendizagem-de-maquina-knn-nearest⟩. Acesso em: 12 jun. 2019.

OCHI, L. S. *Algoritmos Genéticos: Origem e Evolução*. Disponível em: ⟨http://www.sbmac.org.br/bol/bol-2/artigos/satoru/satoru.html⟩. Acesso em: 12 jun. 2019.

VINICIUS. *Classificação usando kNN*. Disponível em: ⟨https://www.monolitonimbus.com.br/classificacao-usando-knn/⟩. Acesso em: 12 jun. 2019.