Reconhecimento de Dígitos Cursivos

Carlos Rodrigo Cordeiro Garcia[1]
Departamento de Informática e Estatística, DEINFO
Universidade Federal Rural de Pernambuco, UFRPE
[1] carlos.rodrigogarcia@ufrpe.br

Giovanni Paolo Santos de Carvalho[2]
Departamento de Informática e Estatística, DEINFO
Universidade Federal Rural de Pernambuco, UFRPE
[2] giovanni.paolo@ufrpe.br

***Abstract*— In this paper we discuss a few approaches to handwritten characters recognition using k-Nearest Neighbors, suport vector machines and convolutional networks. The study proposed is based upon the MNIST digits database - a dataset containing handwritten numbers from 0 to 9 with 60,000 training samples and 10,000 test samples. An overview of the features of the database is also included, properties such as attributes distribution, means, standard deviations and others alike. An off-line handwritten recognition system proposal is discussed along with methods to improve its classification. Image processing and feature extraction are one of the proposed methods for the classification of the characters. The proposed recognition system may have various applications, such as ZIP code recognition in letters or number recognition from bank accounts and paychecks.**

***Index Terms*— Handwriting character recognition; Feature extraction; Support Vector Machines; Convolutional networks;**

I. Introdução

Reconhecimento de escrita é a capacidade de um computador receber e interpretar uma entrada manuscrita inteligível a partir de fontes tais como documentos em papel, imagens, telas sensíveis ao toque e outros dispositivos. Este problema tem sido uma das áreas de pesquisa mais fascinantes e desafiadores em campo de processamento de imagens e reconhecimento de padrões nos últimos anos.[1] Em geral, podemos classificar o problema de reconhecimento de escrita em dois tipos: off-line e on-line. No método off-line, normalmente, a escrita é capturada por algum tipo de scanner e ela é disponibilizada para o classificador como uma imagem, já no on-line os movimentos de uma caneta podem ser sentidos por algum tipo de sensor para detectar-se o que está sendo escrito, como por exemplo, usando telas de computador sensíveis à toques e canetas.[2] A abordagem on-line tem se mostrado superior ao método off-line no reconhecimento de escrita, porém os sistemas off-line também mostraram resultados bastante satisfatórios principalmente com a utilização de redes neurais e redes recorrentes. Várias aplicações como processamento de cheques bancários, reconhecimento de CEP e leitura de documentos escritos à mão precisam de um sistema que consiga reconhecer escrita em um documento normal. Dessa forma reconhecimento de escrita off-line continua sendo uma área de pesquisa bastante relevante para pesquisadores que possam explorar novas técnicas e melhorar tanto na acurácia, generalização e complexidade desses sistemas.

II. Trabalhos Relacionados

Em [2] são consideradas algumas abordagens para aprendizado e classificação online e offline para reconhecimento de escrita cursiva. Em [3], uma aplicação de um MLP visando performance ótima é articulada. Vale também ressaltar o trabalho em [8] que apresenta os melhores métodos até agora de reconhecimento de caracteres off-line. Por fim o trabalho apresentado em [9] que é um artigo em forma de overview da história e dos métodos para reconhecimento de escrita em geral.

III. Base de Dados

A base de dados MNIST é uma das bases mais famosas para o treinamento e teste de classificadores para o problema de reconhecimento de escrita, originalmente criada por Yann LeCun, Courant Institute, NYU com contribuição de Corinna Cortes, Google Labs, New York e Christopher J.C. Burges, Microsoft Research, Redmond. A base de dados MNIST contém um conjunto de treinamento com 60,000 exemplos, e um conjunto de teste com 10,000 exemplos de dígitos entre 0 e 9. Este conjunto é um subconjunto de uma base de dados maior que é disponibilizado pelo NIST(National Institute of Standards and Technology) e pode ser encontrado em[10]. O conjunto de treino contém exemplos de cerca de 250 escritores, o conjunto de escritores da base de teste e de treinamento conjunto são disjuntos. Cada dígito é normalizado e centralizado numa imagem em níveis de cinza com tamanho 28 x 28, resultando numa imagem de 784 pixels, ou seja, existem 784 atributos ao total nessa base de dados que variam do pixel 1 ao 785 como características. Alguns exemplos são mostrados na Figura 1.

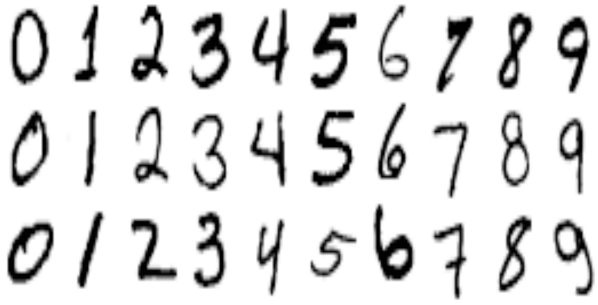


Figura 1: base de dados MNIST.

IV. Análise dos dados

Parte da análise dos dados é feita utilizando-se representações gráficas que correspondem à distribuição dos dados a serem usados pelo algoritmo. Base de dados que contenham padrões com poucas características são mais fáceis de serem visualizados e dessa forma verificados. Assim, se tivéssemos uma base de dados com 2 atributos poderíamos lê-los num gráfico 2D onde o eixo X representa o atributo 1 e o eixo Y representa o atributo 2, podemos assim aumentar a dimensionalidade do problema e continuar a representá-lo num gráfico, por exemplo, para 3 atributos teríamos um espaço com 3 dimensões, onde cada vértice representaria 1 dos atributos do problema. Porém o quanto mais a dimensionalidade cresce mais difícil fica para poder aplicar essa técnica de visualização. Uma solução para esse problema é tentar representar dados de uma dimensão alta em um espaço menor. Normalmente tentamos utilizar uma visualização num espaço de 2 dimensões. Para alcançar esse resultado desejado utilizaremos redução não-linear de dimensionalidade(nonlinear dimensionality reduction), ou manifold learning[4]. Vamos utilizar uma técnica chamada t-SNE que visualiza essas alta dimensões mapeando cada datapoint para um gráfico com duas ou três dimensões[5].

O resultado da aplicação do t-SNE na base de dados MNIST é mostrado na Figura 2.

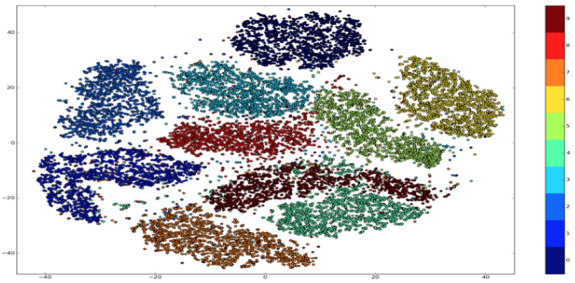


Figura 2: Classes da base MNIST representada em 2D

Dados como média e desvio padrão também são fornecidos na tabela 1, vale lembrar que como a imagem tem 784 pixels em tons de cinza, cada pixel representa uma característica que tem valores mínimos e máximos de 0 e 255, respectivamente. Pelo fato de termos 784 atributos apenas os 5 mais relevantes, selecionados usando um dos métodos de seleção de atributos demonstrados no próximo item, de toda a base de treinamento foram incluídos na tabela abaixo.

Atributo	Média	Desvio Padrão
Pixel 379	113.7	114.3
Pixel 407	132.6	113.9
Pixel 351	90.2	109.3
Pixel 435	138.1	112.5
Pixel 410	129.8	113.3

Média e Desvio Padrão dos 5 atributos mais relevantes

V. Seleção de atributos

O método de seleção de atributos é um dos principais fatores para obter-se uma boa taxa de classificação e um classificador robusto para reconhecimento de caracteres, sem mencionar a redução custo computacional que é alcançado quando pode-se extrair características de uma imagem, reduzindo a dimensionalidade, mas mantendo sua representatividade. Existem ainda casos em que a classificação melhora substancialmente depois da seleção, provando ser um passo extremamente importante para o reconhecimento de caracteres.

Foram utilizados 3 métricas de seleção de atributos: ganho de informação, razão de ganho e correlação. Os resultados obtidos foram bastante parecidos entre essas 3 abordagens, principalmente entre ganho de informação e razão de ganho. Os atributos que tiveram as maiores taxas de relevância, ou seja, aqueles que tem mais informação sobre essa base de dados foram os pixel que estavam na região central da imagem. Analogamente, os atributos com menor importância foram os primeiros, os laterais e últimos pixels, ou seja, atributos que estavam localizados nas extremidades da imagem.

O que faz sentido e comprovou aquilo que nós criamos como modelo para seleção. Os pixels centrais contêm mais informação, dado que esta é a região onde há maior presença de dados; influenciado pelo fato da base possuir imagens centralizadas e pela natureza da escrita de uma pessoa que geralmente preenche espaços centrais de uma determinada região. Já as regiões das extremidades, usualmente, não contêm muitas informações para um determinado caractere pois essa região tende a não ser preenchida quando estamos escrevendo algo num documento.

Pela quantidade de atributos ser muito grande, uma abordagem interativa é proposta para selecionar os atributos. As características são ordenadas por relevância e um limiar local θ que representa o ganho de informação que o píxel tem em relação ao seu vizinho de menor relevância, e um limiar global β que define o máximo de características em % que podem ser extraídas.

Finalmente, o total 784 atributos foi reduzido para 150. Isto equivale à 19% do número inicial de atributos, ou seja, o espaço de características foi diminuido em 81%, enquanto a *acurácia* sofreu um pequeno impacto de 2.84 pontos percentuais após a redução de atributos utilizando uma SVM para classificação.

VI. Algoritmos de apredizagem de máquina utilizados

Foram utilizados 3 algoritmos de apredizagem de máquina pra solucionar este problema, foram eles o *k-nearest neighbors* (k-NN), *Support Vector Machine* (SVM) e uma *Convolutional neural network* (ConvNet).

O classificador que obteve os melhores resultados foi a rede neural convolucionária, chegando à uma acurácia de 98.61%. Os resultados dos classificadores são mostrados nas tabelas abaixo:

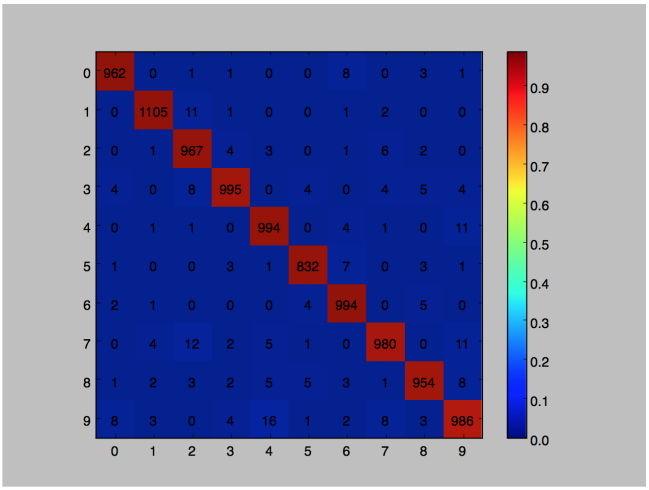
Dígito	Acurácia	Recall	F1-Score	Support
0	0.98	0.99	0.98	976
1	0.99	0.99	0.99	1120
2	0.96	0.98	0.97	984
3	0.98	0.97	0.98	1024
4	0.97	0.98	0.98	1012
5	0.98	0.98	0.98	848
6	0.97	0.99	0.98	1006
7	0.98	0.97	0.97	1015
8	0.98	0.97	0.97	984
9	0.96	0.96	0.96	1031
Total	0.98	0.98	0.98	10000

Acurácia, F-Measure para a ConvNet

Dígito	Acurácia	Recall	F1-Score	Support
0	0.96	0.99	0.97	980
1	0.97	0.99	0.98	1135
2	0.94	0.93	0.94	1032
3	0.93	0.94	0.93	1010
4	0.93	0.95	0.94	982
5	0.93	0.91	0.92	892
6	0.95	0.96	0.96	958
7	0.95	0.93	0.94	1028
8	0.94	0.91	0.93	974
9	0.94	0.91	0.93	1009
Total	0.94	0.94	0.94	10000

Acurácia, F-Measure para o SVM

Para o ConvNet que teve 98% de acurácia, obtivemos esta matrix de confusão, podemos observar melhor com esta representação a distribuicão dos erros do classificador por dígito.



Matrix de confusão da ConvNet com 98% de acurácia

Uma parte importante de para uma boa classificação e consequentemente a geração de um bom classificador é a redução da dimensionalidade através da extração dos atributos, e como mencionado anteriormente, foi possível reduzir de 784 para 150 atributos sem muita perda de classificação. Para validar esta extracão foi-se utilizado validação cruzada. Resultados da extração dos atributos são mostrados abaixo:

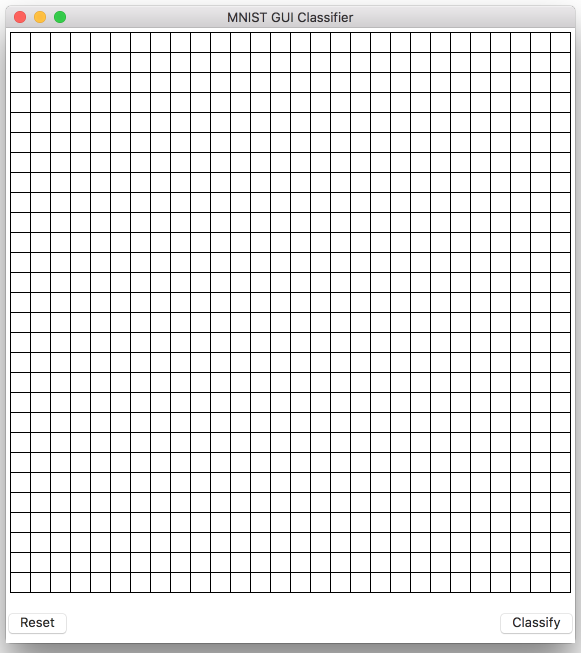
Classificador	Sem extração	Com extração	Folds
ConvNet	98.61%	95.45%	5
SVM	92.92%	90.08%	10
k-NN	89.17%	86.76%	10

Extração de atributos e validação cruzada

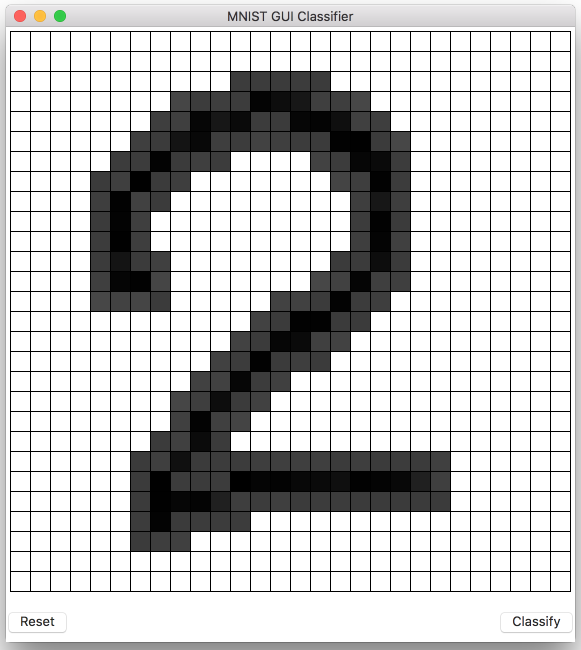
Pode-se notar que a taxa de classificação continuou competitiva para todos os algoritmos testados mesmo diminuindo mais de 80% das características inicias da imagem.

VII. Leitor de Entrada de Usuário

Juntamente à este projeto foi desenvolvida uma ferramente para auxiliar os testes da qualidade dos modelos gerados. Trata-se de uma simples interface visual com um canvas dividido em 28x8 que reconhece desenhos feitos com o mouse para simular escrita num papel ou tela sensível à toque.



Ao clicar e arrastar o mouse sobre o canvas, os espaços vão sendo preenchidos e gerando um vetor de 784 valores correspondente aos níveis de cinza. O espaço diretamente abaixo do mouse é preenchido com uma tonalidade mais escura, equanto os 8 vizinhos são colateralmente preenchidos com tonalidades mais claras.



Embora uma aproximação bem rudimentar, os resultados obtidos foram satisfatórios, sendo correto na maioria das vezes. Contudo, são problemas notáveis a centralização e normalização do tamanho dos dígitos - problemas estes que são tratados na base de dados original do MNIST - bem como a intensidade dos níveis de cinza, que foram estimados e geradores aleatoriamente dentro de intervalos, [230, 255] para os pixels diretamente abaixo do mouse e [200, 210] para os vizinhos. Uma abordagem melhor visaria mudanças de níveis que representassem melhor a pressão de uma escrita levando em consideração o tempo que o risco levou pra ser feito, bem como procurando centralizar e ajustar o dígito.

VIII. Trabalhos Futuros

Visa-se futuramente expandir o reconhecimento para um grupo mais abrangente de caracteres e não apenas números. Também há uma possibilidade de reconhecer letras em frases e não apenas caracteres avulsos, utilizando métodos de segmentação para poder separar uma palavra em letras que possam ser reconhecidas pelo algoritmo a ser implementado.

IX. Conclusão

O estudo desenvolvido ao longo desta pesquisa foi proveitoso, visto que exploramos diversos aspectos do problema e otimizamos ele para para satisfazer melhores patamares de exatidão e eficiência. Atingimos ainda taxas de acertos altíssimas, como 98.61% utilizando ConvNet, uma porcentagem muito satisfatória para o cenário do problema. Além do fato de termos dimuido o espaço de características do problema em até 81% do original mantendo uma taxa de acerto competitiva perdendo apenas 3.16% no pior caso.

References

[1] S. Mori, C.Y. Suen, K. Kamamoto, , *On-line and off-line handwriting recognition: a comprehensive survey.*, Proc. of IEEE, vol. 80, pp. 1029-1058, July 1992.
[2] Réjean Plamondon, Sargur N. Srihari, *Historical review of OCR research and development*, IEEE Trans Pattern Anal Mach Intell.
[3] Anita Pal, Dayashankar Singh, "Handwritten English Character Recognition Using Neural Network"International Journal of Computer Science and Communication.
[4] Michael Gashler, Dan Ventura, e Tony Martinez, “Manifold Learning by Graduated Optimization” IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART B: CYBERNETICS, VOL. 41, NO. 6, DECEMBER 2011.
[5] Laurens van der Maaten, Geoffrey Hinton “Visualizing Data using t-SNE” Journal of Machine Learning Research 9 (2008) 2579-2605.
[6] G. Cybenko, “Approximation by superpositions of a sigmoidal function,” Math. Contr., Signals, Syst., vol. 2, pp. 303–314, 1989.
[7] K. Hornik, M. Stinchcombe and H. White (1989). Multilayer feedforward networks are universal approximators. Neural Networks, 2, 359-366.
[8] B. Verma, M. Blumenstein, S. Kulkarni, “Recent achievements in offline handwriting recognition system”
[9] Homayoon S.M. Beigi, “An Overview of Handwriting Recognition”
[10] MNIST dataset, http://yann.lecun.com/exdb/mnist/
[11] MNIST formato ARFF, http://axon.cs.byu.edu/data/mnist/