Comparação de Desempenho de Redes Aplicado a Reconhecimento de Dígitos Desenhados à Mão

José Geraldo Fernandes
Escola de Engenharia
Universidade Federal de Minas Gerais
Belo Horizonte, Brasil
josegeraldof@ufmg.br

***Resumo*—Este trabalho avalia a performance de diferentes classificadores para o reconhecimento de dígitos desenhados à mão de um subconjunto do MNIST. Utilizou-se redes SVM, extração de características com PCA e CNN.**

I. Introdução

O MNIST [1] é um extenso conjunto de imagens de dígitos desenhados à mão, comumente utilizado em aplicações de *machine learning*. Cada imagem corresponde a um quadrado de 28 *pixels* de uma escala de cinza. O subconjunto utilizado neste trabalho contém apenas os dígitos 1, 5, 6 e 7, como ilustra a Figura 1, e 13017 amostras.

Figura 1. Amostras do subconjunto utilizado.

Cada amostra é portanto tratada como um vetor único de 784 dimensões, um atributo corresponde a um pixel da imagem orientado horizontalmente. A base é quase equilibrada, sua distribuição é como na Figura 2.

Para validação também utilizou-se um conjunto de 4000 amostras sem classificação para propósito de uma competição.

II. Implementação

Para a implementação dos modelos utilizou-se pacotes do R e Python de interesse.

O *kernlab* [2] é um pacote de *machine learning* baseado em *kernel* com vários métodos de classificação, regressão e *clustering*, foi utilizado para a implementação do SVM.

A rede CNN foi implementada em Python por conta da biblioteca Keras [3], que oferece uma interface poderosa e intuitiva para o desenvolvimento de redes neurais artificiais.

A implementação de todos os modelos deste trabalho está disponível no nosso repositório [4] no *GitHub*.

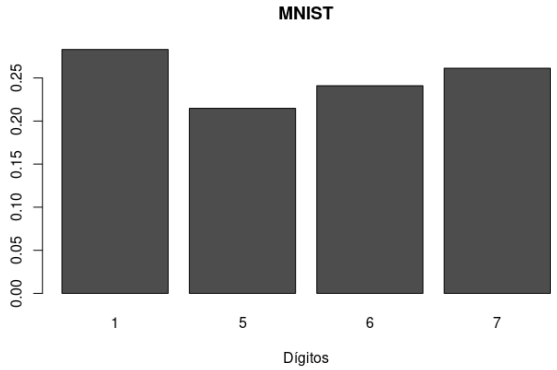

Figura 2. Distribuição dos dígitos.

SVM

SVM - *Support Vector Machines* é uma forma eficiente e econômica de representar curvas complexas em espaços de alta dimensão [5]. São baseados nos limites das classes no espaço a partir da máxima separação das observações mais próximas.

Para um problema binário, e seu conjunto de dados $D = \{(\boldsymbol{x_i}, y_i)|x_i \in \mathbb{R}^{\mathrm{m}}, \mathrm{y_i}\}_i$, a regra de decisão é como na Equação 1 [6].

$$\hat{y}(\boldsymbol{x}) = sign(\sum \hat{\alpha} y_i K(\boldsymbol{x}_i, \boldsymbol{x}) + \hat{b}) \tag{1}$$

Sendo $K(\boldsymbol{x}_i, \boldsymbol{x})$ a função de *kernel*. Neste trabalho utilizou-se a função de base radial gaussiana, RBF.

PCA

PCA - *Principal Component Analysis* é uma forma não supervisionada de extração de características importantes a partir da variância dos dados, reduzindo também a complexidade espacial [7].

O método é simples, basta transformar o espaço na direção dos autovetores calculados das amostras. Cada autovalor carrega uma parcela da variância e pode-se selecionar apenas um subconjunto representativo.

CNN

Redes convolucionais têm o poder de treinar redes de múltiplas camadas capturando mapeamentos complexos, não

lineares e de alta dimensão [8]. Por esse motivo, são candidatas naturais para a solução deste tipo de problema, e, em verdade, estado da arte no tópico, uma simples inspeção nos classificadores referenciados pela equipe do MNIST [1] afirma esse ponto.

Neste trabalho, foi utilizada uma arquitetura não tão moderna mas suficiente para o problema, a LeNet-5, adaptada para quatro classes, utilizada também no artigo do MNIST [8].

A arquitetura LeNet-5 corresponde de sete camadas: duas convolucionais; duas de subamostragem; e, duas totalmente conectadas.

A primeira camada, de convolução, corresponde de 6 filtros 3x3 com ativação ReLU, aqui a entrada de 28x28 foi preenchida com zeros para corresponder a entrada 32x32 da arquitetura.

Segundo, uma camada de subamostragem com valor médio e tamanho 2x2. A terceira e quarta são uma repetição das primeiras, com exceção de 16 filtros de convolução em vez de 6.

Finalmente, as últimas camadas totalmente conectadas com 120 e 84 unidades com, também, ativação ReLU. A saída é *one-hot* nas quatro classes. A Figura 3 [8] mostra um esquema da arquitetura.

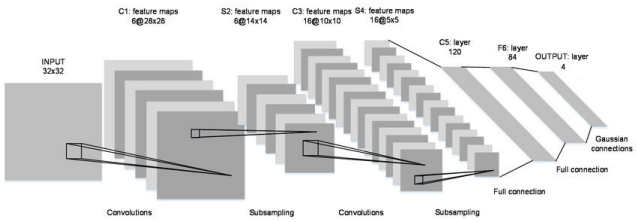

Figura 3. Arquitetura LeNet-5, apenas quatro saídas.

III. Experimentos

Os experimentos seguiram o padrão para treinamento de redes artificiais. Para cada modelo selecionou-se os parâmetros pelo método *k-fold* de validação cruzada, utilizando dez partições.

Uma etapa adicional foi realizada por motivo da competição, a classificação das amostras sem rótulo. Para tanto, selecionou-se o melhor modelo das dez partições, diga-se, aquele com maior acurácia de teste, para inferência.

Sobre esse conjunto, é importante destacar algumas hipóteses não comprovadas que foram aproveitadas neste trabalho.

Número um, as amostras estão em ordem de classificação, isto é, o primeiro quarto delas são da classe 1 e assim por diante. Assim sendo, é possível avaliar a acurácia da inferência.

Número dois, as amostras são representativas do conjunto rotulado. Em outras palavras, ambos conjuntos têm a mesma distribuição e diversidade. Com exceção, é claro, do equilíbrio total das classes, por consequência da hipótese número um.

Essas hipóteses foram levantadas por uma inspeção manual e superficial das amostras não rotuladas.

IV. Resultados

Segue a acurácia média e da validação dos modelos, como na Tabela I.

Utilizou-se uma SVM com *kernel* RBF a partir dos dados puros e, também, de um pré-processamento por PCA, com 40 atributos. A CNN utilizada foi a arquitetura LeNet-5.

Tabela I
Acurácia

Modelo	*k-folds*	Validação*
SVM	0.992 ± 0.002	0.992
SVM-PCA	0.992 ± 0.002	0.151
LeNet-5	0.995 ± 0.002	0.995

V. Discussão

Como esperado, o desempenho da CNN foi superior. Mesmo assim, é surpreendente o quanto a SVM não perde por muito, apesar de a inclusão de novas classes (dígitos que foram suprimidos) poder aumentar essa diferença. A equipe do MNIST mantém uma tabela de referência [1] de acurácia, os valores encontrados não foram distoantes.

Apesar disso, o que salta os olhos é a acurácia de validação do modelo SVM com PCA. É possível uma inspeção visual das duas primeiras componentes do PCA, como na Figura 4 e 5.

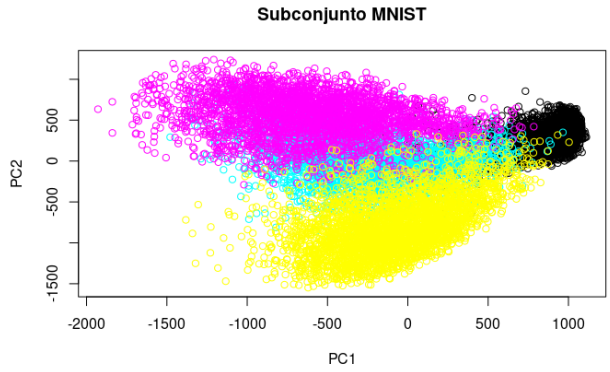

Figura 4. Amostras rotuladas, em preto 1, em azul 5, em rosa 6 e em amarelo 7.

Este resultado ameaça as hipóteses levantadas já que a distribuição das classes é muito diferente na representação em duas dimensões.

Mesmo assim, ainda há motivos para manter a confiança. Primeiro, os números dos outros modelos não mostraram o mesmo fenômeno e estão de acordo com a acurácia de teste. Segundo, a distribuição de classes pode variar a representação em componentes PCA.

Para aprofundar no problema, um novo PCA foi calculado com a união dos dois conjuntos. De fato, a distribuição dos dados não rotulados foi impactada, como na Figura 6.

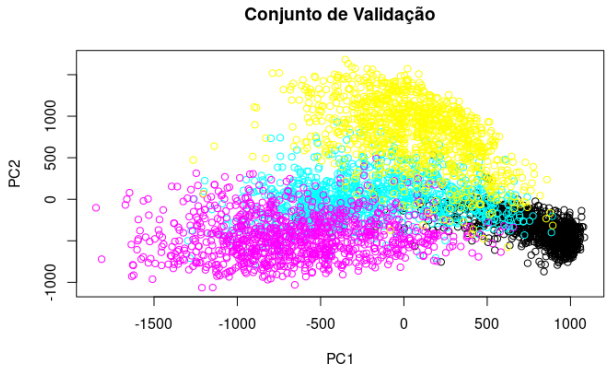

Figura 5. Amostras não rotuladas.

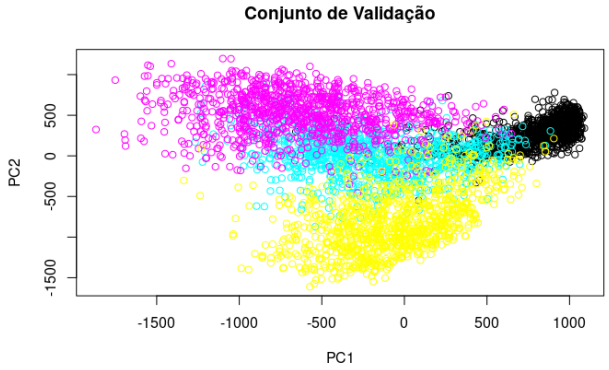

Figura 6. Amostras não rotuladas, PCA inclusivo.

Contudo, independente, a acurácia de validação não alcançou o esperado pela baixo acerto dos dígitos 5 e 6, como na Tabela II.

Tabela II
ACURÁCIA POR CLASSE

Geral	1	5	6	7
0.58	0.86	0.10	0.54	0.83

VI. CONCLUSÕES

Demonstrou-se a superioridade das redes CNN para o problema de classificação de manuscritos, em especial pelo seu poder de generalização espacial. Mesmo assim, o modelo SVM se mostrou muito eficiente, apesar de tratado com um conjunto reduzido do problema.

Sobre a representação PCA, partindo apenas dos resultados no conjunto rotulado é muito proveitosa, já que mantém o desempenho, reduz a dimensionalidade e o tempo de treinamento.

Sobre as hipóteses do conjunto não rotulado é particularmente perigoso o tratamento, já que as evidências são fracas e conflitantes. De forma forçada, pode-se comentar que caso a hipótese número dois seja falsa, ou seja, o conjunto não rotulado seja, entre outros, de baixa qualidade, caligrafia ruim, ou sofrido alguma variação espacial, o método PCA não é capaz de generalizar a informação como fez o SVM e CNN.

REFERÊNCIAS

[1] Y. Lecun, L. Bottou, Y. Bengio, P. Haffner. MNIST handwritten digit database. http://yann.lecun.com/exdb/mnist/
[2] Alexandros Karatzoglou (2019). Kernel-Based Machine Learning Lab. https://CRAN.R-project.org/package=kernlab
[3] F. Chollet e outros (2015). Keras. https://github.com/keras-team/keras
[4] José Geraldo Fernandes. MNIST. https://github.com/josegfer/mnist/
[5] V. N. Vapnik (1995). The Nature of Statistical Learning Theory.
[6] B. Clarke, E. Fokoué e H. Zhang (2009). Principals and Theory for Data Mining and Machine Learning. http://dx.doi.org/10.1007/978-0-387-98135-2
[7] R. Duda, P. Hart e D. Stork (2000). Pattern Classification.
[8] Y. Lecun, L. Bottou, Y. Bengio, P. Haffner (1998). Gradient-based learning applied to document recognition. https://doi.org/10.1109/5.726791