

# ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO



## **Fatoração de Matrizes para Sistemas de Classificação de Dígitos Manuscritos**

### **Machine Learning**

MAP3121 - Cálculo Numérico

Alessandro Brugnera Silva	10334040
Marcelo Hiromi Soares Arita	10256936

São Paulo, 19 de maio de 2019

## Sumário

<b>Arquivos</b>	<b>3</b>
<b>Programas de processamento</b>	<b>4</b>
<b>Funções Auxiliares</b>	<b>5</b>
Tarefa 1	5
Utilização	5
Funções auxiliares	5
Exemplo	5
Discussão sobre a tarefa 1	7
Tarefa 1A	7
Tarefa 1B	8
Tarefa 1C	9
Tarefa 1D	10
Tarefa 2	11
Utilização	11
Funções Auxiliares	11
Exemplo:	11
<b>Tarefa Principal</b>	<b>13</b>
Leitura dos dados mnist	13
Treinamento do algoritmo	13
Procedimento	13
Eficiência	13
Teste com menos amostras de imagens	13
Funções Auxiliares	13
Classificação dos dígitos	14
Comparação dos resultados (ndig = 5300)	14
Comentários sobre a comparação:	16
<b>Considerações finais</b>	<b>18</b>
Opinião do aluno Marcelo	18

# Arquivos

- Programas em Python
  - EP2.py
    - Arquivo principal, roda e chama as outras funções em outros arquivos.
  - Tarefas
    - Tarefa1A.py
    - Tarefa1B.py
    - Tarefa1C.py
    - Tarefa1D.py
  - CriaA.py
    - A partir dos dados\_mnist indexa as matrizes A em uma array numpy.
  - Funções auxiliares
    - Tarefa2.py
      - A partir de p e a matriz A, retorna a matriz W refinada.
  - ResolveSimult.py
  - RotGivens.py
  - Triangularizar.py
- Matrizes
  - A
    - .txt
      - Matrizes fornecidas para processamento
    - .npy
      - Matrizes já indexadas pela biblioteca numpy. Criando um arquivo externo para agilizar a leitura da matriz
  - $W_{\text{dig}} p_p$ 
    - .npy
      - Matrizes de aprendizado do dígito (0-9) e p (5, 10 ou 15)
- Dados gerados com extensão npy
  - Wdpn.npy - conjunto de matrizes W treinados
- CSVs gerados
  - TempoCria.csv
    - Tempos em cada coluna 0 1 2 3 4 5 6 7 8 9
    - Linhas p 5 10 15
  - TempoTesta.csv
    - Acerto de cada dígito (10 numeros)
    - Tempos de cada P (1 n[umero)
    - 0 (separação para outro p.
- Eficiencia.pdf
  - Tabela com tempos e acertos.

# Programas de processamento

- PyCharm
  - Utilizado para aprendizado e geração de W.
  - Gerou os tempos e acertos nos testes com diferentes  $n_{digs}$ .

# Funções Auxiliares

## • Tarefa 1

### ○ Utilização

- Método de resolução de MMQ, por meio da decomposição QR
- A função recebe inteiros:  $m$ ,  $n$  e  $p$  e arrays  $W_{n \times p}$  e  $A_{n \times m}$
- A função retorna uma matriz  $H_{p \times m}$ , tal que:  $WH = A$ , através da resolução de  $m$  sistemas simultâneos

### ○ Funções auxiliares

- Rotgivers<sup>1</sup>( $W$ ,  $n$ ,  $m$ ,  $i$ ,  $j$ ,  $c$ ,  $s$ )
  - Aplica a rotação de Givens à matriz  $W$  na forma vetorial.
- Triangularizar ( $W$ ,  $A$ )
  - Aplicação de Rotgivers às matrizes  $W$  e  $A$ , retornando uma matriz triangular superior  $R$  e uma matriz  $A$  também aplicado Rotgivers.
  -
- ResolvSimult ( $m$ ,  $R$ ,  $A$ )
  - Dadas as matrizes  $R$  e  $A$  retornadas pela função Triangularizar, a função retorna uma outra matriz  $H$ , tal que  $WH = A$ , por meio de resolução de  $m$  sistemas simultâneos.

### ○ Exemplo

Seja a matriz  $W_{3 \times 3}$  dada por :

	2	1	0
	1	2	1
	0	1	2

Queremos eliminar o elemento 1, da posição  $w_{2 \times 1}$ . Ao aplicar a função Rotgivers, obteremos uma nova matriz  $W$  dada por:

2,236067977	1,788854382	0,447213595
0	1,341640786	0,894427191
0	1	2

---

<sup>1</sup> A função Rotgivers foi fornecida pelo IME, disponível em [https://www.ime.usp.br/~map3121/2019/map3121/programas/vector\\_oper.py](https://www.ime.usp.br/~map3121/2019/map3121/programas/vector_oper.py)  
Foi utilizada a função de número 5, presente no endereço acima.

Se continuarmos aplicando a função Rotgivens à matriz W, ao final de todo o processo obteremos uma matriz triangular superior dada por:

2,236067977	1,788854382	0,447213595
0	1,673320053	1,912365775
0	0	1,069044968

- Observe que:
  - A função Rotgivens altera apenas os valores contidos nas linhas  $i$  e  $j$  da matriz. Lembrando  $i = j - 1$  sendo  $j$  o *índice da linha* do elemento ao qual se pretende eliminar.
  - Embora não colocamos nos exemplos acima, a função Rotgivens pode ser aplicado tanto à matriz W quando à matriz A. No entanto, a triangularização não ocorre nesta última.

- Discussão sobre a tarefa 1

- Tarefa 1A

Segue abaixo, a lista com as soluções encontradas pelo algoritmo

X		X		X		X	
Xi	valor	Xi	valor	Xi	valor	Xi	valor
1	0,492307692	21	0,338461538	41	0,184615385	61	0,030769231
2	0,015384615	22	0,169230769	42	0,323076923	62	0,476923077
3	0,476923077	23	0,323076923	43	0,169230769	63	0,015384615
4	0,030769231	24	0,184615385	44	0,338461538	64	0,492307692
5	0,461538462	25	0,307692308	45	0,153846154		
6	0,046153846	26	0,2	46	0,353846154		
7	0,446153846	27	0,292307692	47	0,138461538		
8	0,061538462	28	0,215384615	48	0,369230769		
9	0,430769231	29	0,276923077	49	0,123076923		
10	0,076923077	30	0,230769231	50	0,384615385		
11	0,415384615	31	0,261538462	51	0,107692308		
12	0,092307692	32	0,246153846	52	0,4		
13	0,4	33	0,246153846	53	0,092307692		
14	0,107692308	34	0,261538462	54	0,415384615		
15	0,384615385	35	0,230769231	55	0,076923077		
16	0,123076923	36	0,276923077	56	0,430769231		
17	0,369230769	37	0,215384615	57	0,061538462		
18	0,138461538	38	0,292307692	58	0,446153846		
19	0,353846154	39	0,2	59	0,046153846		
20	0,153846154	40	0,307692308	60	0,461538462		

Tabela com valores das soluções para o item (a) da tarefa 1

**Erro quadrático calculado: 0**

(Sistema determinado,  $n = m = 64$ ).

- Tarefa 1B

Segue abaixo, a lista com as soluções encontradas pelo algoritmo

X	
Xi	valor
1	53,47625739
2	-44,04108632
3	-41,9750347
4	-47,05464148
5	-29,6864054
6	83,95542005
7	45,29171909
8	55,58268495
9	10,24267243
10	103,0373931
11	-70,39304087
12	-52,88496122
13	-50,24051459
14	-25,14322932
15	92,07034982
16	207,1678027
17	283,8197017
18	0
19	0
20	0

Repare que as últimas 3 (  $n - m$  ) linhas da solução possuem valores 0. Isso ocorre pelo fato da matriz triangular R também possuir as mesmas linhas nulas!

Tabela com valores das soluções para o item (b) da tarefa 1

**Erro quadrático calculado:** 48.84142247062892

(Sistema sobredeterminado  $n = 20 > m = 17$ ).



○ Tarefa 1C

Segue abaixo, a lista com as soluções encontradas pelo algoritmo

H				H				H				H			
Hi	0	1	2	Hi	0	1	2	Hi	0	1	2	Hi	0	1	2
1	0,492307692	-0,492307692	-1,476923077	21	0,338461538	-0,338461538	-1,015384615	41	0,184615385	-0,184615385	-0,553846154	61	0,030769231	-0,030769231	-0,092307692
2	0,015384615	0,984615385	1,953846154	22	0,169230769	10,83076923	21,49230769	42	0,323076923	20,67692308	41,03076923	62	0,476923077	30,52307692	60,56923077
3	0,476923077	-0,476923077	-1,430769231	23	0,323076923	-0,323076923	-0,969230769	43	0,169230769	-0,169230769	-0,507692308	63	0,015384615	-0,015384615	-0,046153846
4	0,030769231	1,969230769	3,907692308	24	0,184615385	11,81538462	23,44615385	44	0,338461538	21,66153846	42,98461538	64	0,492307692	31,50769231	62,52307692
5	0,461538462	-0,461538462	-1,384615385	25	0,307692308	-0,307692308	-0,923076923	45	0,153846154	-0,153846154	-0,461538462				
6	0,046153846	2,953846154	5,861538462	26	0,2	12,8	25,4	46	0,353846154	22,64615385	44,93846154				
7	0,446153846	-0,446153846	-1,338461538	27	0,292307692	-0,292307692	-0,876923077	47	0,138461538	-0,138461538	-0,415384615				
8	0,061538462	3,938461538	7,815384615	28	0,215384615	13,78461538	27,35384615	48	0,369230769	23,63076923	46,89230769				
9	0,430769231	-0,430769231	-1,292307692	29	0,276923077	-0,276923077	-0,830769231	49	0,123076923	-0,123076923	-0,369230769				
10	0,076923077	4,923076923	9,769230769	30	0,230769231	14,76923077	29,30769231	50	0,384615385	24,61538462	48,84615385				
11	0,415384615	-0,415384615	-1,246153846	31	0,261538462	-0,261538462	-0,784615385	51	0,107692308	-0,107692308	-0,323076923				
12	0,092307692	5,907692308	11,72307692	32	0,246153846	15,75384615	31,26153846	52	0,4	25,6	50,8				
13	0,4	-0,4	-1,2	33	0,246153846	-0,246153846	-0,738461538	53	0,092307692	-0,092307692	-0,276923077				
14	0,107692308	6,892307692	13,67692308	34	0,261538462	16,73846154	33,21538462	54	0,415384615	26,58461538	52,75384615				
15	0,384615385	-0,384615385	-1,153846154	35	0,230769231	-0,230769231	-0,692307692	55	0,076923077	-0,076923077	-0,230769231				
16	0,123076923	7,876923077	15,63076923	36	0,276923077	17,72307692	35,16923077	56	0,430769231	27,56923077	54,70769231				
17	0,369230769	-0,369230769	-1,107692308	37	0,215384615	-0,215384615	-0,646153846	57	0,061538462	-0,061538462	-0,184615385				
18	0,138461538	8,861538462	17,58461538	38	0,292307692	18,70769231	37,12307692	58	0,446153846	28,55384615	56,66153846				
19	0,353846154	-0,353846154	-1,061538462	39	0,2	-0,2	-0,6	59	0,046153846	-0,046153846	-0,138461538				
20	0,153846154	9,846153846	19,53846154	40	0,307692308	19,69230769	39,07692308	60	0,461538462	29,53846154	58,61538462				

Tabela com os valores da matriz H calculado para o item (c) da tarefa 1

**Erro quadrático calculado: 0**

(Sistema determinado. Isto significa que  $A - W \cdot H = 0$ , com W e A originais).

○ Tarefa 1D

Segue abaixo, 3 tabelas:

- H calculado
- A original
- $M = W \cdot H$ , com W original

H			
Hi	0	1	2
1	2,881550631	53,47625739	104,0709641
2	-1,833762532	-44,04108632	-86,24841011
3	-1,513989039	-41,9750347	-82,43608036
4	-1,52190762	-47,05464148	-92,58737534
5	-0,453794614	-29,6864054	-58,91901618
6	5,856698891	83,95542005	162,0541412
7	3,421928907	45,29171909	87,16150928
8	3,656562195	55,58268495	107,5088077
9	1,203684541	10,24267243	19,28166031
10	6,125342488	103,0373931	199,9494437
11	-2,479713964	-70,39304087	-138,3063678
12	-1,477931471	-52,88496122	-104,291991
13	-1,203902584	-50,24051459	-99,2771266
14	0,12841469	-25,14322932	-50,41487334
15	6,501588998	92,07034982	177,6391106
16	11,49105599	207,1678027	402,8445494
17	14,58052582	283,8197017	553,0588776

A			
Ai	0	1	2
1	1	0	-1
2	1	1	1
3	1	2	3
4	1	3	5
5	1	4	7
6	1	5	9
7	1	6	11
8	1	7	13
9	1	8	15
10	1	9	17
11	1	10	19
12	1	11	21
13	1	12	23
14	1	13	25
15	1	14	27
16	1	15	29
17	1	16	31
18	1	17	33
19	1	18	35
20	1	19	37

M			
Mi	0	1	2
1	0,988770524	-0,236905459	-1,462581442
2	0,907681189	-0,801023183	-2,509727555
3	0,993101049	1,863626043	2,734151037
4	1,152515901	6,042638924	10,93276195
5	1,107797783	6,111076526	11,11435527
6	0,927500548	3,547593851	6,167687155
7	0,968014747	5,364248945	9,760483143
8	0,982534944	6,654229206	12,32592347
9	0,989457337	7,791871175	14,59428501
10	0,965163869	8,274561028	15,58395819
11	0,801861483	6,138001188	11,47414089
12	0,997975445	10,96001574	20,92205603
13	1,277102966	17,52623858	33,7753742
14	1,176793163	16,46048747	31,74418179
15	0,881532644	11,62741644	22,37330024
16	0,94973366	14,00119542	27,05265717
17	0,9734249	15,47406296	29,97470102
18	0,984369878	16,69157701	32,39878413
19	0,951576536	16,9923156	33,03305467
20	0,733330492	13,80296068	26,87259087

Tabela com os valores da matriz H calculado para o item (d) da tarefa 1

Matriz A original

Matriz M, tal que  $M = W \cdot H$  (com W original)

**Erro quadrático calculado:** 541.1721016854622

Para fins de comparação, colocamos as duas tabelas A e M. Note que nesse caso o existe um erro quadrático que se origina pelo "afastamento" notável entre essas duas matrizes. Como resolvemos um sistema sobredeterminado, era esperado um erro quadrático como este.

## ● Tarefa 2

### ○ Utilização

- A função basicamente deve aplicar a fatoração WH em uma matriz A com um parâmetro p escolhido.
- Por meio de iterações, a partir de uma matriz  $W_{n \times p}$  aleatória, as matrizes  $W_{n \times p}$  e  $H_{p \times m}$  são refinadas até a precisão de  $10^{-5}$  do erro  $A-W*H$ .
- Utiliza-se as funções da tarefa 1 para resolver o sistema sobredeterminado, utilizando o método MMQ, criado pela fato de  $n \geq m$ .
- Ao final, gera-se matrizes W e H, cujo produto é suficientemente próximo de A para utilização na rede neural.

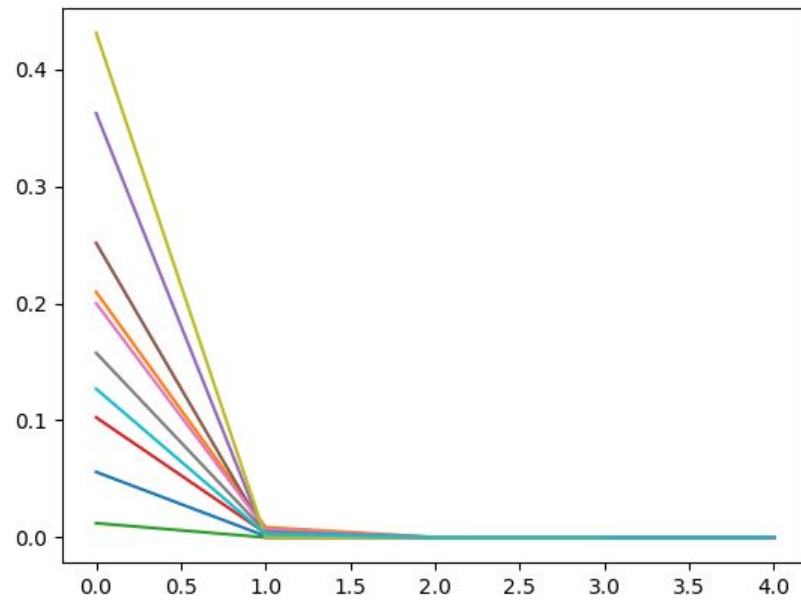
### ○ Funções Auxiliares

- ResolveSimult.py
  - A função recebe sistemas simultâneos sobredeterminados e retorna a solução de  $A*x=B$ .
  - Utilizada para resolver os sistemas ( $W*H=A$  e  $H^t*W^t=A^t$ ) nos processos iterativos.
  - A função é criada na Tarefa 1.
- NormaColuna
  - Normaliza a matriz.
- Diferenca
  - Calcula  $E=A-W*H$ .

### ○ Exemplo:

$$A = \begin{bmatrix} 3/10 & 3/5 & 0 \\ 1/2 & 0 & 1 \\ 4/10 & 4/5 & 0 \end{bmatrix} W = \begin{bmatrix} 3/5 & 0 \\ 0 & 1 \\ 4/5 & 0 \end{bmatrix} H = \begin{bmatrix} 1/2 & 1 & 0 \\ 1/2 & 0 & 1 \end{bmatrix}$$

- Aplicando o método em A com  $p=2$ .
- Com m, n baixos a fatoração é rapidamente alcançada; com cerca de 4 iterações. Porém com grande precisão vide:
  - Gráfico do erro convergindo para 0.
  - Matrizes próximas das corretas.
- Gráfico de Erro Quadrático em função da iteração, em 10 tentativas com W randômico.



- Média termo a termo das 10 matrizes W e H.
- A matrizes se aproximam muito das matrizes exatas. Com erro tendendo a 0.

$$W = \begin{bmatrix} 6.00062980 * 10 - 01 & 6.30300154 * 10 - 05 \\ 5.22358886 * 10 - 06 & 1.00000501 * 10 + 00 \\ 8.00083974 * 10 - 01 & 8.40400205 * 10 - 05 \end{bmatrix}$$

$$H = \begin{bmatrix} 0.49968492 & 1.000000002 & 0. \\ 0.49998443 & 0. & 1.000000022 \end{bmatrix}$$

# Tarefa Principal

## Classificação de dígitos manuscritos

### ● Leitura dos dados mnist

- A partir dos dados pré-processados, que representam uma matriz de cada pixel das imagens, foram criadas as matrizes  $A_{\text{dig}}$  que foram salvas utilizando a função save da biblioteca numpy.
- Com as matrizes já indexadas pelo numpy, foi facilitado o aprendizado de  $W_{\text{dig}}$ , possibilitando que o código rodasse mais rapidamente com menos memória consumida. Além disso caso um erro ocorresse, o código podia ser reiniciado sem a necessidade de indexar  $W$  novamente.

### ● Treinamento do algoritmo

- Procedimento
  - Nesta parte todas as outras funções foram utilizadas.
  - Partindo das matrizes  $A_{\text{dig}}$  já indexadas com todas as imagens de teste, e com 4 parâmetros de  $p$  (5, 10, 15, 20); o algoritmo da tarefa 2 foi utilizado.
  - Dessa forma 40 matrizes  $W_{\text{dig}p_p}$  foram geradas e armazenadas com a função save do numpy.
  - A função time foi utilizada para avaliar o tempo de aprendizado de cada dígito com os 4 “p”s.
- Eficiência
  - PDF Eficiência em anexo.

### ● Funções Auxiliares

- Diferenca\_c ( $C, j$ )
  - Dada uma matriz  $C$ , a função retorna a norma euclidiana de sua coluna  $j$ .
- Classificar ( $A_{\text{test}}$ , Gabarito,  $p$ )
  - Dada a matriz de teste  $A_{\text{test}}$ , chama-se as funções da tarefa 1 para resolver o sistema simultâneo  $W_d H = A_{\text{test}}$ , por meio de MMQ.
  - Cria-se uma lista com as normas euclidianas de cada matriz  $M$ , tal que  $M_d = A_{\text{test}} - W_d H$  com  $H$  calculado na etapa anterior.
  - Para cada  $M_d$ , armazenamos a sua norma euclidiana de cada coluna. Note que  $M_d$  tem dimensão  $10 \times n_{\text{test}}$ .
  - Finalmente, buscamos o elemento que possui a menor norma euclidiana para cada coluna. A linha que intercepta essa coluna é aquela que representa o dígito.
  - A função retorna a lista de possíveis dígitos, a taxa de acertos total e individual para cada dígito.

- Taxa (C,G)
  - Dada uma lista com dígitos e uma lista de gabarito, calcula a taxa total de dígitos corretos da lista em relação ao gabarito (no caso, test\_index).
- Frequencia (C,G)
  - Dada uma lista com dígitos e uma lista de gabarito, retorna a taxa de acerto individual para cada dígito.

## ● Classificação dos dígitos

- Procedimento
  - Nesta etapa, foram utilizadas todas as funções da tarefa 1 e 2.
  - Após a fase anterior de treinamento, utilizamos os arrays  $W_{dig}P_p$  treinados para a execução de testes.
  - Os dados para os testes foram obtidos do arquivo mnist, contendo 10000 imagens de teste.
  - Para cada p, foi utilizada a função Classificar (A\_test, Gabarito, p)

## ● Comparação dos resultados (ndig = 5300)

- p = 5

Resultados para p = 5 são:

A taxa de acertos é de: 91.63 %  
 A frequencia de acertos para o dígito 0 é: 97.75510204081633 %  
 A frequencia de acertos para o dígito 1 é: 99.29515418502203 %  
 A frequencia de acertos para o dígito 2 é: 89.82558139534885 %  
 A frequencia de acertos para o dígito 3 é: 92.67326732673268 %  
 A frequencia de acertos para o dígito 4 é: 87.78004073319755 %  
 A frequencia de acertos para o dígito 5 é: 87.5560538116592 %  
 A frequencia de acertos para o dígito 6 é: 96.65970772442589 %  
 A frequencia de acertos para o dígito 7 é: 89.2023346303502 %  
 A frequencia de acertos para o dígito 8 é: 88.7063655030801 %  
 A frequencia de acertos para o dígito 9 é: 85.72844400396433 %

	0	1	2	3
0	0	980	958	97.7551
1	1	1135	1127	99.2952
2	2	1032	927	89.8256
3	3	1010	936	92.6733
4	4	982	862	87.78
5	5	892	781	87.5561
6	6	958	926	96.6597
7	7	1028	917	89.2023
8	8	974	864	88.7064
9	9	1009	865	85.7284

- p = 10

Resultados para p = 10 são:  
A taxa de acertos é de: 93.4799999999999 %  
A frequencia de acertos para o dígito 0 é: 98.26530612244898 %  
A frequencia de acertos para o dígito 1 é: 99.64757709251101 %  
A frequencia de acertos para o dígito 2 é: 91.76356589147287 %  
A frequencia de acertos para o dígito 3 é: 93.66336633663367 %  
A frequencia de acertos para o dígito 4 é: 92.76985743380855 %  
A frequencia de acertos para o dígito 5 é: 89.01345291479821 %  
A frequencia de acertos para o dígito 6 é: 96.86847599164928 %  
A frequencia de acertos para o dígito 7 é: 91.73151750972762 %  
A frequencia de acertos para o dígito 8 é: 88.60369609856262 %  
A frequencia de acertos para o dígito 9 é: 91.37760158572844 %

	0	1	2	3
0	0	980	963	98.2653
1	1	1135	1131	99.6476
2	2	1032	947	91.7636
3	3	1010	946	93.6634
4	4	982	911	92.7699
5	5	892	794	89.0135
6	6	958	928	96.8685
7	7	1028	943	91.7315
8	8	974	863	88.6037
9	9	1009	922	91.3776

○ p = 15

Resultados para p = 15 são:  
A taxa de acertos é de: 93.89 %  
A frequencia de acertos para o dígito 0 é: 98.77551020408163 %  
A frequencia de acertos para o dígito 1 é: 99.20704845814979 %  
A frequencia de acertos para o dígito 2 é: 91.56976744186046 %  
A frequencia de acertos para o dígito 3 é: 93.86138613861385 %  
A frequencia de acertos para o dígito 4 é: 95.01018329938901 %  
A frequencia de acertos para o dígito 5 é: 91.25560538116592 %  
A frequencia de acertos para o dígito 6 é: 96.76409185803759 %  
A frequencia de acertos para o dígito 7 é: 90.95330739299611 %  
A frequencia de acertos para o dígito 8 é: 88.50102669404517 %  
A frequencia de acertos para o dígito 9 é: 92.26957383548068 %

	0	1	2	3
0	0	980	968	98.7755
1	1	1135	1126	99.207
2	2	1032	945	91.5698
3	3	1010	948	93.8614
4	4	982	933	95.0102
5	5	892	814	91.2556
6	6	958	927	96.7641
7	7	1028	935	90.9533
8	8	974	862	88.501
9	9	1009	931	92.2696

○ p = 20

Resultados para  $p = 20$  são:

A taxa de acertos é de: 94.08 %

A frequência de acertos para o dígito 0 é: 98.67346938775509 %

A frequência de acertos para o dígito 1 é: 98.94273127753304 %

A frequência de acertos para o dígito 2 é: 92.05426356589147 %

A frequência de acertos para o dígito 3 é: 91.78217821782178 %

A frequência de acertos para o dígito 4 é: 94.09368635437882 %

A frequência de acertos para o dígito 5 é: 90.91928251121077 %

A frequência de acertos para o dígito 6 é: 97.4947807933194 %

A frequência de acertos para o dígito 7 é: 93.19066147859922 %

A frequência de acertos para o dígito 8 é: 90.14373716632443 %

A frequência de acertos para o dígito 9 é: 92.76511397423192 %

	0	1	2	3
0	0	980	967	98.6735
1	1	1135	1123	98.9427
2	2	1032	950	92.0543
3	3	1010	927	91.7822
4	4	982	924	94.0937
5	5	892	811	90.9193
6	6	958	934	97.4948
7	7	1028	958	93.1907
8	8	974	878	90.1437
9	9	1009	936	92.7651

## Comentários sobre a comparação:

Notamos que a taxa de acerto total cresce com o aumento do parâmetro  $p$ , independentemente do valor de  $ndig$ . Se olharmos com mais detalhes, isso não ocorre de maneira regular entre cada dígito individual. Alguns dígitos sofreram um decaimento de sua taxa de acerto quando  $p$  foi aumentado. Por exemplo, repare que o dígito de número 3 para  $p = 15$  teve um acerto de 93,86%. Para o mesmo dígito, com  $p = 20$  notamos que a taxa caiu para 91,78%. O mesmo fenômeno ocorreu para os dígitos 0, 1, 4 e 5. Para os demais, houve um acréscimo na sua taxa de acerto - o que elevou globalmente a totalidade de dígitos corretos.

Observamos também que a variação dos parâmetros  $ndig$ , isto é, o espaço de amostras de imagens para treinarmos as matrizes  $W$ s, também influenciou nos resultados de acertos. Mantendo-se  $p$  constante, mas variando esse valor, chegamos em um aumento direto: quanto maior o valor de  $ndig$ , maior também é a taxa de acerto total e discriminado, para cada algoritmo testado.

Assim, esperava-se que o teste que obteria maior êxito em termos de acerto total fosse para  $ndig = 5300$  e  $p = 15$ . Entretanto, para  $ndig = 4000$  e  $p = 15$ , o êxito foi maior. Para o primeiro, tivemos um acerto de 93,89% enquanto que para este último, 93,98%. Uma diferença pequena, mas que precisa ser destacada. Nos demais, à grosso modo, podemos dizer que o aumento de amostras de imagens para o treinamento e de  $p$  permitem uma melhora na taxa de acerto global.



# Considerações finais

## Opinião do aluno Marcelo

Gostei bastante do EP, principalmente porque ele nos trouxe uma aplicação bastante prática e de altíssima relevância na esfera da engenharia. No entanto o enunciado está confuso em alguns trechos, sem uma padronização ao fazer referência a um mesmo objeto etc. Muitas vezes fiquei confuso por não entender se uma determinada informação era um input pelo usuário ou se era para ser colocado diretamente ao código.

Senti também que a disciplina de introdução a Python, ministrada no primeiro ano foi muito mal dada, visto que tanto as aulas quanto os Eps dessa disciplina possui um foco muito simples e supérfluo. Faz se necessário, portanto, uma mudança no conteúdo ensinado, nos Eps dados aos alunos (Eps que não sejam joguinhos de cobra, por exemplo).

Por fim, deixo meu elogio à equipe que estruturou esse material, pois além de ter aumentado bastante a minha habilidade de programar, também me interessei mais por outras aplicações correlacionadas ao tema central deste Ep. Muito obrigado.