



Universidade Federal de Uberlândia  
Faculdade de Engenharia Elétrica



## Aprendizagem de Máquina

# Trabalho 04 - Reconhecimento de dígitos usando Perceptrons

### **Docente:**

Prof. Keiji Yamanaka

### **Discente:**

Augusto Soares Porto - 12121ECP016

Uberlândia  
5 de setembro de 2024

## Sumário

### Sumário

<b>1</b>	<b>Introdução</b>	<b>2</b>
<b>2</b>	<b>Código realizado</b>	<b>2</b>
2.1	reconhecimento_de_digitos_por_perceptron.py . . . . .	2
2.2	digitos20x20.py . . . . .	6
<b>3</b>	<b>Saída do código realizado</b>	<b>16</b>
3.1	Teste 1 . . . . .	17
3.2	Teste 2 . . . . .	19
3.3	Teste 3 . . . . .	21
<b>4</b>	<b>Referências</b>	<b>23</b>

# 1 Introdução

Perceptrons são um tipo de rede neural artificial capaz de aprender a classificar corretamente as letras a partir de um conjunto de dados de treinamento, analisando o impacto de diferentes parâmetros, como o peso das conexões e o valor de bias, na eficácia do reconhecimento.

Este relatório foca no estudo e implementação de um algoritmo de perceptron para o reconhecimento de algarismos na fonte Times New Roman, o qual pode ser treinado para diferenciar os 10 algarismos indo-arábicos, cada uma representado por uma matriz de pixels 20x20.

## 2 Código realizado

## 2.1 reconhecimento de dígitos por perceptron.py

```

1 from digitos_20x20 import digitos
2
3
4 def main():
5     # Treinar o perceptron
6     taxa_aprendizado = 0.020 # adotado
7     w, b = algoritmo_perceptron(digitos, taxa_aprendizado)
8
9     # Testar com uma entrada nova
10    x = [
11        -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1,
12        -1, -1, -1, -1,
13        -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1,
14        -1, -1, -1, -1,
15        -1, -1, -1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
16        1, 1, 1, -1,
17        -1, -1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
18        1, 1, -1, -1,
19        -1, 1, 1, 1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, 1, 1,
20        1, -1, -1, -1,
21        -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, 1, 1, 1,
22        -1, -1, -1, -1,
23    ]
24
25    # Testar com uma entrada nova
26    y = 0
27
28    # Testar com uma entrada nova
29    y = 0
30
31    # Testar com uma entrada nova
32    y = 0
33
34    # Testar com uma entrada nova
35    y = 0
36
37    # Testar com uma entrada nova
38    y = 0
39
40    # Testar com uma entrada nova
41    y = 0
42
43    # Testar com uma entrada nova
44    y = 0
45
46    # Testar com uma entrada nova
47    y = 0
48
49    # Testar com uma entrada nova
50    y = 0
51
52    # Testar com uma entrada nova
53    y = 0
54
55    # Testar com uma entrada nova
56    y = 0
57
58    # Testar com uma entrada nova
59    y = 0
60
61    # Testar com uma entrada nova
62    y = 0
63
64    # Testar com uma entrada nova
65    y = 0
66
67    # Testar com uma entrada nova
68    y = 0
69
70    # Testar com uma entrada nova
71    y = 0
72
73    # Testar com uma entrada nova
74    y = 0
75
76    # Testar com uma entrada nova
77    y = 0
78
79    # Testar com uma entrada nova
80    y = 0
81
82    # Testar com uma entrada nova
83    y = 0
84
85    # Testar com uma entrada nova
86    y = 0
87
88    # Testar com uma entrada nova
89    y = 0
90
91    # Testar com uma entrada nova
92    y = 0
93
94    # Testar com uma entrada nova
95    y = 0
96
97    # Testar com uma entrada nova
98    y = 0
99
100   # Testar com uma entrada nova
101   y = 0
102
103   # Testar com uma entrada nova
104   y = 0
105
106   # Testar com uma entrada nova
107   y = 0
108
109   # Testar com uma entrada nova
110   y = 0
111
112   # Testar com uma entrada nova
113   y = 0
114
115   # Testar com uma entrada nova
116   y = 0
117
118   # Testar com uma entrada nova
119   y = 0
120
121   # Testar com uma entrada nova
122   y = 0
123
124   # Testar com uma entrada nova
125   y = 0
126
127   # Testar com uma entrada nova
128   y = 0
129
130   # Testar com uma entrada nova
131   y = 0
132
133   # Testar com uma entrada nova
134   y = 0
135
136   # Testar com uma entrada nova
137   y = 0
138
139   # Testar com uma entrada nova
140   y = 0
141
142   # Testar com uma entrada nova
143   y = 0
144
145   # Testar com uma entrada nova
146   y = 0
147
148   # Testar com uma entrada nova
149   y = 0
150
151   # Testar com uma entrada nova
152   y = 0
153
154   # Testar com uma entrada nova
155   y = 0
156
157   # Testar com uma entrada nova
158   y = 0
159
160   # Testar com uma entrada nova
161   y = 0
162
163   # Testar com uma entrada nova
164   y = 0
165
166   # Testar com uma entrada nova
167   y = 0
168
169   # Testar com uma entrada nova
170   y = 0
171
172   # Testar com uma entrada nova
173   y = 0
174
175   # Testar com uma entrada nova
176   y = 0
177
178   # Testar com uma entrada nova
179   y = 0
180
181   # Testar com uma entrada nova
182   y = 0
183
184   # Testar com uma entrada nova
185   y = 0
186
187   # Testar com uma entrada nova
188   y = 0
189
190   # Testar com uma entrada nova
191   y = 0
192
193   # Testar com uma entrada nova
194   y = 0
195
196   # Testar com uma entrada nova
197   y = 0
198
199   # Testar com uma entrada nova
200   y = 0
201
202   # Testar com uma entrada nova
203   y = 0
204
205   # Testar com uma entrada nova
206   y = 0
207
208   # Testar com uma entrada nova
209   y = 0
210
211   # Testar com uma entrada nova
212   y = 0
213
214   # Testar com uma entrada nova
215   y = 0
216
217   # Testar com uma entrada nova
218   y = 0
219
220   # Testar com uma entrada nova
221   y = 0
222
223   # Testar com uma entrada nova
224   y = 0
225
226   # Testar com uma entrada nova
227   y = 0
228
229   # Testar com uma entrada nova
230   y = 0
231
232   # Testar com uma entrada nova
233   y = 0
234
235   # Testar com uma entrada nova
236   y = 0
237
238   # Testar com uma entrada nova
239   y = 0
240
241   # Testar com uma entrada nova
242   y = 0
243
244   # Testar com uma entrada nova
245   y = 0
246
247   # Testar com uma entrada nova
248   y = 0
249
250   # Testar com uma entrada nova
251   y = 0
252
253   # Testar com uma entrada nova
254   y = 0
255
256   # Testar com uma entrada nova
257   y = 0
258
259   # Testar com uma entrada nova
260   y = 0
261
262   # Testar com uma entrada nova
263   y = 0
264
265   # Testar com uma entrada nova
266   y = 0
267
268   # Testar com uma entrada nova
269   y = 0
270
271   # Testar com uma entrada nova
272   y = 0
273
274   # Testar com uma entrada nova
275   y = 0
276
277   # Testar com uma entrada nova
278   y = 0
279
280   # Testar com uma entrada nova
281   y = 0
282
283   # Testar com uma entrada nova
284   y = 0
285
286   # Testar com uma entrada nova
287   y = 0
288
289   # Testar com uma entrada nova
290   y = 0
291
292   # Testar com uma entrada nova
293   y = 0
294
295   # Testar com uma entrada nova
296   y = 0
297
298   # Testar com uma entrada nova
299   y = 0
300
301   # Testar com uma entrada nova
302   y = 0
303
304   # Testar com uma entrada nova
305   y = 0
306
307   # Testar com uma entrada nova
308   y = 0
309
310   # Testar com uma entrada nova
311   y = 0
312
313   # Testar com uma entrada nova
314   y = 0
315
316   # Testar com uma entrada nova
317   y = 0
318
319   # Testar com uma entrada nova
320   y = 0
321
322   # Testar com uma entrada nova
323   y = 0
324
325   # Testar com uma entrada nova
326   y = 0
327
328   # Testar com uma entrada nova
329   y = 0
330
331   # Testar com uma entrada nova
332   y = 0
333
334   # Testar com uma entrada nova
335   y = 0
336
337   # Testar com uma entrada nova
338   y = 0
339
340   # Testar com uma entrada nova
341   y = 0
342
343   # Testar com uma entrada nova
344   y = 0
345
346   # Testar com uma entrada nova
347   y = 0
348
349   # Testar com uma entrada nova
350   y = 0
351
352   # Testar com uma entrada nova
353   y = 0
354
355   # Testar com uma entrada nova
356   y = 0
357
358   # Testar com uma entrada nova
359   y = 0
360
361   # Testar com uma entrada nova
362   y = 0
363
364   # Testar com uma entrada nova
365   y = 0
366
367   # Testar com uma entrada nova
368   y = 0
369
370   # Testar com uma entrada nova
371   y = 0
372
373   # Testar com uma entrada nova
374   y = 0
375
376   # Testar com uma entrada nova
377   y = 0
378
379   # Testar com uma entrada nova
380   y = 0
381
382   # Testar com uma entrada nova
383   y = 0
384
385   # Testar com uma entrada nova
386   y = 0
387
388   # Testar com uma entrada nova
389   y = 0
390
391   # Testar com uma entrada nova
392   y = 0
393
394   # Testar com uma entrada nova
395   y = 0
396
397   # Testar com uma entrada nova
398   y = 0
399
400   # Testar com uma entrada nova
401   y = 0
402
403   # Testar com uma entrada nova
404   y = 0
405
406   # Testar com uma entrada nova
407   y = 0
408
409   # Testar com uma entrada nova
410   y = 0
411
412   # Testar com uma entrada nova
413   y = 0
414
415   # Testar com uma entrada nova
416   y = 0
417
418   # Testar com uma entrada nova
419   y = 0
420
421   # Testar com uma entrada nova
422   y = 0
423
424   # Testar com uma entrada nova
425   y = 0
426
427   # Testar com uma entrada nova
428   y = 0
429
430   # Testar com uma entrada nova
431   y = 0
432
433   # Testar com uma entrada nova
434   y = 0
435
436   # Testar com uma entrada nova
437   y = 0
438
439   # Testar com uma entrada nova
440   y = 0
441
442   # Testar com uma entrada nova
443   y = 0
444
445   # Testar com uma entrada nova
446   y = 0
447
448   # Testar com uma entrada nova
449   y = 0
450
451   # Testar com uma entrada nova
452   y = 0
453
454   # Testar com uma entrada nova
455   y = 0
456
457   # Testar com uma entrada nova
458   y = 0
459
460   # Testar com uma entrada nova
461   y = 0
462
463   # Testar com uma entrada
```

```

17     -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, 1, 1, 1,
        -1, -1, -1, -1, -1,
18     -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, 1, 1, 1, -1,
        -1, -1, -1, -1, -1,
19     -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, 1, 1, 1, -1, -1,
        -1, -1, -1, -1, -1,
20     -1, -1, -1, -1, -1, -1, -1, -1, -1, 1, 1, 1, -1, -1, -1,
        -1, -1, -1, -1, -1,
21     -1, -1, -1, -1, -1, -1, -1, -1, 1, 1, 1, -1, -1, -1, -1,
        -1, -1, -1, -1, -1,
22     -1, -1, -1, -1, -1, -1, -1, 1, 1, 1, -1, -1, -1, -1, -1,
        -1, -1, -1, -1, -1,
23     -1, -1, -1, -1, -1, -1, 1, 1, 1, -1, -1, -1, -1, -1, -1,
        -1, -1, -1, -1, -1,
24     -1, -1, -1, -1, -1, 1, 1, 1, -1, -1, -1, -1, -1, -1, -1,
        -1, -1, -1, -1, -1,
25     -1, -1, -1, -1, 1, 1, 1, -1, -1, -1, -1, -1, -1, -1, -1,
        -1, -1, -1, -1, -1,
26     -1, -1, -1, 1, 1, 1, -1, -1, -1, -1, -1, -1, -1, -1, -1,
        -1, -1, -1, -1, -1,
27     -1, -1, 1, 1, 1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1,
        -1, -1, -1, -1, -1,
28     -1, 1, 1, 1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1,
        -1, -1, -1, -1, -1,
29     -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1,
        -1, -1, -1, -1,
30     -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1,
        -1, -1, -1, -1,
31     -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1,
        -1, -1, -1, -1
32 ]
33
34
35 y = teste_perceptron(x, w, b)
36
37 # Exibir o resultado do perceptron
38 estimativa = None
39
40 for i, digito in enumerate(digitos.keys()):
41     print(f"Saída para a dígito {digito}: y = {y[i]}")
42     if y[i] == 1:

```

```
43         estimativa = digitos[digito]
44
45     print("Entrada:")
46     print_digito(x)
47
48     if estimativa is None:
49         print("Não foi possível identificar a dígito")
50     else:
51         print("Saída estimada:")
52         print_digito(estimativa)
53
54
55 # Função para imprimir a digito
56 def print_digito(digito):
57     for i in range(20):
58         for j in range(20):
59             print('#' if digito[i*20 + j] == 1 else ' ', end='')
60         print()
61
62
63 # Função de ativação
64 def ativacao(yliq):
65     if yliq >= 0: # adotei = 0
66         return 1
67     else:
68         return -1
69
70
71 # Função para treinar o perceptron
72 def algoritmo_perceptron(digitos, taxa_aprendizado):
73     num_digitos = len(digitos)
74     w = [[0] * 400 for i in range(num_digitos)]
75     b = [0] * num_digitos
76     alfa = taxa_aprendizado
77
78     while True:
79         pesos_mudaram = False
80
81         for i, (digito, valores) in enumerate(digitos.items()):
82             x = valores
83
```

```
84         t = []
85         for j in range(num_digitos):
86             if j == i:
87                 t.append(1)
88             else:
89                 t.append(-1)
90
91         for j in range(num_digitos):
92             yliq = sum(w[j][k] * x[k] for k in range(400)) + b[
93                 j]
94             y = ativacao(yliq)
95
96             if y != t[j]:
97                 pesos_mudaram = True
98                 for k in range(400):
99                     w[j][k] += alfa * x[k] * t[j]
100                     b[j] += alfa * t[j]
101
102         if not pesos_mudaram:
103             break
104
105     return w, b
106
107 # Função para testar o perceptron
108 def teste_perceptron(x, w, b):
109     y = []
110     for i in range(len(w)):
111         yliq = sum(w[i][k] * x[k] for k in range(400)) + b[i]
112         y.append(ativacao(yliq))
113     return y
114
115
116 if __name__ == '__main__':
117     main()
```

Listing 1 – Exemplo de código Python

## 2.2 digitos<sub>20x20</sub>.py

```

1 # # Definição dos dígitos com fonte Times New Roman
2 digitos = {
3     '1': [
4         -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, 1, -1, -1, -1, -1,
5         -1, -1, -1, -1,
6         -1, -1, -1, -1, -1, -1, -1, -1, -1, 1, 1, 1, -1, -1, -1, -1,
7         -1, -1, -1, -1,
8         -1, -1, -1, -1, -1, -1, -1, 1, 1, 1, 1, 1, -1, -1, -1, -1, -1,
9         -1, -1, -1, -1,
10        -1, -1, -1, -1, -1, -1, -1, -1, -1, 1, 1, 1, -1, -1, -1, -1,
11        -1, -1, -1, -1,
12        -1, -1, -1, -1, -1, -1, -1, -1, -1, 1, 1, 1, -1, -1, -1, -1,
13        -1, -1, -1, -1,
14        -1, -1, -1, -1, -1, -1, -1, -1, -1, 1, 1, 1, -1, -1, -1, -1,
15        -1, -1, -1, -1,
16        -1, -1, -1, -1, -1, -1, -1, -1, -1, 1, 1, 1, -1, -1, -1, -1,
17        -1, -1, -1, -1,
18        -1, -1, -1, -1, -1, -1, -1, -1, -1, 1, 1, 1, -1, -1, -1, -1,
19        -1, -1, -1, -1,
20        -1, -1, -1, -1, -1, -1, -1, -1, -1, 1, 1, 1, -1, -1, -1, -1,
21        -1, -1, -1, -1,
22        -1, -1, -1, -1, -1, -1, -1, -1, 1, 1, 1, 1, 1, -1, -1, -1, -1,
23        -1, -1, -1, -1,

```





```

44 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
    1, -1, -1, -1,
45 -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1,
    -1, -1, -1, -1,
46 -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1,
    -1, -1, -1, -1
47 ],
48
49 '3': [
50 -1, -1, -1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, -1, -1, -1, -1,
    -1, -1, -1, -1,
51 -1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, -1, -1, -1,
    -1, -1, -1, -1,
52 -1, 1, 1, 1, 1, -1, -1, -1, -1, -1, -1, 1, 1, 1, -1, -1,
    -1, -1, -1, -1,
53 -1, 1, 1, -1, -1, -1, -1, -1, -1, -1, -1, -1, 1, 1, -1, -1,
    -1, -1, -1, -1,
54 -1, 1, 1, -1, -1, -1, -1, -1, -1, -1, -1, -1, 1, 1, -1, -1,
    -1, -1, -1, -1,
55 -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, 1, 1, -1, -1,
    -1, -1, -1, -1,
56 -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, 1, 1, 1, -1, -1,
    -1, -1, -1, -1,
57 -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, 1, 1, 1, -1, -1, -1,
    -1, -1, -1, -1,
58 -1, -1, -1, -1, -1, -1, -1, 1, 1, 1, 1, 1, -1, -1, -1, -1, -1,
    -1, -1, -1,
59 -1, -1, -1, -1, 1, 1, 1, 1, 1, 1, 1, 1, 1, -1, -1, -1, -1,
    -1, -1, -1,
60 -1, -1, -1, -1, -1, -1, -1, 1, 1, 1, 1, 1, 1, -1, -1, -1, -1, -1,
    -1, -1, -1,
61 -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, 1, 1, 1, -1, -1, -1,
    -1, -1, -1, -1,
62 -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, 1, 1, 1, -1, -1,
    -1, -1, -1, -1,
63 -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, 1, 1, 1, -1, -1,
    -1, -1, -1, -1,
64 -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, 1, 1, 1, -1,
    -1, -1, -1, -1,
65 -1, 1, 1, 1, -1, -1, -1, -1, -1, -1, -1, -1, 1, 1, 1, -1,
    -1, -1, -1, -1,

```

```

66     -1,  1,  1,  1,  1, -1, -1, -1, -1, -1, -1, -1,  1,  1,  1, -1,
        -1, -1, -1, -1,
67     -1,  1,  1,  1,  1,  1, -1, -1, -1, -1, -1,  1,  1,  1, -1, -1,
        -1, -1, -1, -1,
68     -1,  -1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1, -1, -1, -1,
        -1, -1, -1, -1,
69     -1, -1, -1, -1,  1,  1,  1,  1,  1,  1,  1,  1, -1, -1, -1, -1,
        -1, -1, -1, -1
70 ],
71
72 '4': [
73     -1, -1, -1, -1, -1, -1, -1, -1, -1, -1,  1,  1,  1, -1, -1, -1,
        -1, -1, -1, -1,
74     -1, -1, -1, -1, -1, -1, -1, -1, -1,  1,  1,  1,  1, -1, -1, -1,
        -1, -1, -1, -1,
75     -1, -1, -1, -1, -1, -1, -1, -1,  1, -1,  1,  1,  1, -1, -1, -1,
        -1, -1, -1, -1,
76     -1, -1, -1, -1, -1, -1, -1,  1, -1, -1,  1,  1,  1, -1, -1, -1,
        -1, -1, -1, -1,
77     -1, -1, -1, -1, -1, -1,  1, -1, -1, -1,  1,  1,  1, -1, -1, -1,
        -1, -1, -1, -1,
78     -1, -1, -1, -1, -1,  1, -1, -1, -1, -1,  1,  1,  1, -1, -1, -1,
        -1, -1, -1, -1,
79     -1, -1, -1, -1,  1, -1, -1, -1, -1, -1,  1,  1,  1, -1, -1, -1,
        -1, -1, -1, -1,
80     -1, -1, -1,  1, -1, -1, -1, -1, -1, -1,  1,  1,  1, -1, -1, -1,
        -1, -1, -1, -1,
81     -1, -1,  1, -1, -1, -1, -1, -1, -1, -1,  1,  1,  1, -1, -1, -1,
        -1, -1, -1, -1,
82     -1,  1, -1, -1, -1, -1, -1, -1, -1, -1,  1,  1,  1, -1, -1, -1,
        -1, -1, -1, -1,
83     1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1,
        1,  1, -1, -1,
84     1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1,
        1,  1, -1, -1,
85     -1, -1, -1, -1, -1, -1, -1, -1, -1, -1,  1,  1,  1, -1, -1, -1,
        -1, -1, -1, -1,
86     -1, -1, -1, -1, -1, -1, -1, -1, -1, -1,  1,  1,  1, -1, -1, -1,
        -1, -1, -1, -1,
87     -1, -1, -1, -1, -1, -1, -1, -1, -1, -1,  1,  1,  1, -1, -1, -1,
        -1, -1, -1, -1,

```

```

88     -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, 1, 1, 1, -1, -1, -1,
      -1, -1, -1, -1,
89     -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, 1, 1, 1, -1, -1, -1,
      -1, -1, -1, -1,
90     -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, 1, 1, 1, -1, -1, -1,
      -1, -1, -1, -1,
91     -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, 1, 1, 1, -1, -1, -1,
      -1, -1, -1, -1,
92     -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, 1, 1, 1, -1, -1, -1,
      -1, -1, -1, -1
93 ],
94
95     '5': [
96     -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1,
      -1, -1, -1, -1, -1, -1,
97     -1, -1, -1, -1, -1, -1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
      -1, -1, -1, -1,
98     -1, -1, -1, -1, -1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, -1, -1,
      -1, -1, -1,
99     -1, -1, -1, -1, 1, 1, -1, -1, -1, -1, -1, -1, -1, -1, -1,
      -1, -1, -1, -1, -1,
100    -1, -1, -1, 1, 1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1,
      -1, -1, -1, -1, -1,
101    -1, -1, 1, 1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1,
      -1, -1, -1, -1,
102    -1, 1, 1, 1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1,
      -1, -1, -1, -1,
103    1, 1, 1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1,
      -1, -1, -1, -1,
104    1, 1, 1, 1, 1, 1, 1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1,
      -1, -1, -1,
105    -1, -1, -1, -1, 1, 1, 1, 1, 1, 1, 1, -1, -1, -1, -1, -1, -1,
      -1, -1, -1,
106    -1, -1, -1, -1, -1, -1, -1, 1, 1, 1, 1, 1, -1, -1, -1, -1, -1,
      -1, -1, -1,
107    -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, 1, 1, 1, -1, -1, -1,
      -1, -1, -1, -1,
108    -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, 1, 1, 1, -1, -1,
      -1, -1, -1, -1,
109    -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, 1, 1, 1, -1, -1,
      -1, -1, -1, -1,

```

```

110     -1,  -1,  -1, -1, -1, -1, -1, -1, -1, -1, -1, -1,  1,  1,  1, -1,
        -1, -1, -1, -1,
111     1,  1,  1,  -1, -1, -1, -1, -1, -1, -1, -1, -1,  1,  1,  1, -1,
        -1, -1, -1, -1,
112     1,  1,  1,  1,  -1, -1, -1, -1, -1, -1, -1, -1,  1,  1,  1, -1,
        -1, -1, -1, -1,
113     1,  1,  1,  1,  1,  -1, -1, -1, -1, -1, -1,  1,  1,  1, -1, -1,
        -1, -1, -1, -1,
114     -1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1, -1, -1, -1,
        -1, -1, -1, -1,
115     -1, -1, -1, 1,  1,  1,  1,  1,  1,  1,  1,  1, -1, -1, -1, -1,
        -1, -1, -1, -1
116 ],
117
118 '6': [
119     -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1,
        -1, -1, -1, -1,
120     -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1,  1,  1,  1, -1, -1,
        -1, -1, -1, -1,
121     -1, -1, -1, -1, -1, -1, -1, -1, -1,  1,  1,  1, -1, -1, -1, -1,
        -1, -1, -1, -1,
122     -1, -1, -1, -1, -1, -1, -1,  1,  1,  1, -1, -1, -1, -1, -1,
        -1, -1, -1, -1,
123     -1, -1, -1, -1, -1,  1,  1,  1,  1, -1, -1, -1, -1, -1, -1,
        -1, -1, -1, -1,
124     -1, -1, -1, -1,  1,  1,  1, -1, -1, -1, -1, -1, -1, -1, -1,
        -1, -1, -1, -1,
125     -1, -1, -1,  1,  1,  1, -1, -1, -1, -1, -1, -1, -1, -1, -1,
        -1, -1, -1, -1,
126     -1, -1,  1,  1,  1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1,
        -1, -1, -1, -1,
127     -1,  1,  1,  1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1,
        -1, -1, -1, -1,
128     -1,  1,  1,  1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1,
        -1, -1, -1, -1,
129     1,  1,  1,  1,  -1,  1,  1,  1,  1,  1,  1,  1, -1, -1, -1, -1,
        -1, -1, -1, -1,
130     1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1, -1, -1, -1, -1,
        -1, -1,
131     1,  1,  1,  1,  1, -1, -1, -1, -1, -1, -1, -1,  1,  1,  1, -1, -1,
        -1, -1, -1,

```

```

132 1, 1, 1, 1, -1, -1, -1, -1, -1, -1, -1, -1, -1, 1, 1, 1,
    -1, -1, -1, -1,
133 1, 1, 1, 1, -1, -1, -1, -1, -1, -1, -1, -1, -1, 1, 1, 1,
    -1, -1, -1, -1,
134 -1, 1, 1, 1, -1, -1, -1, -1, -1, -1, -1, -1, -1, 1, 1, 1,
    -1, -1, -1, -1,
135 -1, 1, 1, 1, -1, -1, -1, -1, -1, -1, -1, -1, -1, 1, 1, 1,
    -1, -1, -1, -1,
136 -1, -1, 1, 1, 1, -1, -1, -1, -1, -1, -1, -1, -1, 1, 1,
    1, -1, -1, -1, -1, -1,
137 -1, -1, -1, -1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, -1, -1, -1,
    -1, -1, -1, -1,
138 -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1,
    -1, -1, -1, -1
139 ],
140
141 '7': [
142 -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1,
    -1, -1, -1, -1,
143 -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1,
    -1, -1, -1, -1,
144 -1, -1, -1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
    1, 1, 1, -1,
145 -1, -1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
    1, 1, -1, -1,
146 -1, 1, 1, 1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, 1, 1,
    1, -1, -1, -1,
147 -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, 1, 1, 1,
    -1, -1, -1, -1,
148 -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, 1, 1, 1,
    -1, -1, -1, -1, -1,
149 -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, 1, 1, 1, -1,
    -1, -1, -1, -1, -1,
150 -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, 1, 1, 1, -1, -1,
    -1, -1, -1, -1, -1,
151 -1, -1, -1, -1, -1, -1, -1, -1, -1, 1, 1, 1, -1, -1, -1,
    -1, -1, -1, -1, -1,
152 -1, -1, -1, -1, -1, -1, -1, -1, 1, 1, 1, -1, -1, -1, -1,
    -1, -1, -1, -1, -1,
153 -1, -1, -1, -1, -1, -1, -1, 1, 1, 1, -1, -1, -1, -1, -1,
    -1, -1, -1, -1, -1,

```



```

176 -1, -1, -1, -1, 1, 1, 1, 1, 1, 1, 1, 1, -1, -1, -1, -1,
      -1, -1, -1, -1,
177 -1, -1, -1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, -1, -1, -1,
      -1, -1, -1,
178 -1, -1, 1, 1, 1, -1, -1, -1, -1, -1, -1, -1, 1, 1, 1, -1,
      -1, -1, -1, -1,
179 -1, 1, 1, 1, -1, -1, -1, -1, -1, -1, -1, -1, -1, 1, 1, 1,
      -1, -1, -1, -1,
180 -1, 1, 1, 1, -1, -1, -1, -1, -1, -1, -1, -1, -1, 1, 1, 1,
      1, -1, -1, -1,
181 -1, 1, 1, 1, -1, -1, -1, -1, -1, -1, -1, -1, -1, 1, 1, 1,
      1, -1, -1, -1,
182 -1, 1, 1, 1, -1, -1, -1, -1, -1, -1, -1, -1, -1, 1, 1, 1,
      1, -1, -1, -1,
183 -1, -1, 1, 1, 1, -1, -1, -1, -1, -1, -1, -1, -1, 1, 1,
      1, 1, -1, -1, -1, -1,
184 -1, -1, -1, -1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, -1, -1,
      -1, -1, -1, -1,
185 -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1,
      -1, -1, -1, -1
186 ],
187
188 '9': [
189 -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1,
      -1, -1, -1, -1,
190 -1, -1, -1, -1, -1, -1, -1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
      -1, -1, -1, -1,
191 -1, -1, -1, -1, -1, 1, 1, 1, -1, -1, -1, -1, -1, -1,
      -1, 1, 1, 1, -1, -1,
192 -1, -1, -1, -1, 1, 1, 1, -1, -1, -1, -1, -1, -1, -1, -1,
      1, 1, 1, -1,
193 -1, -1, -1, -1, 1, 1, 1, -1, -1, -1, -1, -1, -1, -1, -1,
      1, 1, 1, 1,
194 -1, -1, -1, -1, 1, 1, 1, -1, -1, -1, -1, -1, -1, -1, -1,
      1, 1, 1, 1,
195 -1, -1, -1, -1, 1, 1, 1, -1, -1, -1, -1, -1, -1, -1, -1,
      1, 1, 1, 1,
196 -1, -1, -1, -1, 1, 1, 1, -1, -1, -1, -1, -1, -1, -1, -1,
      1, 1, 1, 1,
197 -1, -1, -1, -1, -1, 1, 1, 1, -1, -1, -1, -1, -1, -1,
      -1, 1, 1, 1, 1, 1,

```

```

198 -1, -1, -1, -1, -1, -1, -1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
    1, 1, 1, 1,
199 -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1,
    -1, 1, 1, 1, -1,
200 -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1,
    -1, 1, 1, 1, -1,
201 -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, 1,
    1, 1, -1, -1,
202 -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, 1, 1,
    1, -1, -1, -1,
203 -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, 1, 1, 1,
    -1, -1, -1, -1,
204 -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, 1, 1, 1, 1, -1,
    -1, -1, -1, -1,
205 -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, 1, 1, 1, -1, -1, -1,
    -1, -1, -1, -1,
206 -1, -1, -1, -1, -1, -1, -1, -1, 1, 1, 1, 1, -1, -1, -1, -1,
    -1, -1, -1, -1,
207 -1, -1, -1, -1, -1, -1, 1, 1, 1, -1, -1, -1, -1, -1, -1, -1,
    -1, -1, -1, -1,
208 -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1,
    -1, -1, -1, -1,
209 ],
210
211 '0': [
212 -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1,
    -1, -1, -1, -1,
213 -1, -1, -1, -1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, -1, -1, -1,
    -1, -1, -1, -1,
214 -1, -1, 1, 1, 1, -1, -1, -1, -1, -1, -1, -1, -1, 1, 1,
    1, -1, -1, -1, -1, -1,
215 -1, 1, 1, 1, -1, -1, -1, -1, -1, -1, -1, -1, -1, 1, 1, 1,
    -1, -1, -1, -1,
216 -1, 1, 1, 1, -1, -1, -1, -1, -1, -1, -1, -1, -1, 1, 1, 1,
    -1, -1, -1, -1,
217 -1, 1, 1, 1, -1, -1, -1, -1, -1, -1, -1, -1, -1, 1, 1, 1,
    -1, -1, -1, -1,
218 -1, 1, 1, 1, -1, -1, -1, -1, -1, -1, -1, -1, -1, 1, 1, 1,
    -1, -1, -1, -1,
219 -1, 1, 1, 1, -1, -1, -1, -1, -1, -1, -1, -1, -1, 1, 1, 1,
    -1, -1, -1, -1,

```



```

220     -1,  1, 1, 1, -1, -1, -1, -1, -1, -1, -1, -1,  -1,  1,  1,  1,
        -1,  -1, -1, -1,
221     -1,  1, 1, 1, -1, -1, -1, -1, -1, -1, -1, -1,  -1,  1,  1,  1,
        -1,  -1, -1, -1,
222     -1,  1, 1, 1, -1, -1, -1, -1, -1, -1, -1, -1,  -1,  1,  1,  1,
        -1,  -1, -1, -1,
223     -1,  1, 1, 1, -1, -1, -1, -1, -1, -1, -1, -1,  -1,  1,  1,  1,
        -1,  -1, -1, -1,
224     -1,  1, 1, 1, -1, -1, -1, -1, -1, -1, -1, -1,  -1,  1,  1,  1,
        -1,  -1, -1, -1,
225     -1,  1, 1, 1, -1, -1, -1, -1, -1, -1, -1, -1,  -1,  1,  1,  1,
        -1,  -1, -1, -1,
226     -1,  1, 1, 1, -1, -1, -1, -1, -1, -1, -1, -1,  -1,  1,  1,  1,
        -1,  -1, -1, -1,
227     -1,  1, 1, 1, -1, -1, -1, -1, -1, -1, -1, -1,  -1,  1,  1,  1,
        -1,  -1, -1, -1,
228     -1,  1, 1, 1, -1, -1, -1, -1, -1, -1, -1, -1,  -1,  1,  1,  1,
        -1, -1, -1, -1,
229     -1, -1,  1,  1,  1,  -1,  -1,  -1,  -1,  -1,  -1,  -1,  -1,  1,  1,
        1, -1, -1, -1, -1, -1,
230     -1, -1, -1,  -1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1, -1, -1, -1,
        -1, -1, -1, -1,
231     -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1,
        -1, -1, -1, -1
232 ],
233
234 }
```

Listing 2 – Exemplo de código Python

### 3 Saída do código realizado

Vale ressaltar que a saída não corresponde a matriz com os 26 neurônios, mas sim a saída ao neurônio correspondente a letra determinada.



```

21  -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1,
    -1, -1, -1, -1,
22  -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1,
    -1, -1, -1, -1
23  ]

```

```

1  Saída para a dígito 1: y = -1
2  Saída para a dígito 2: y = -1
3  Saída para a dígito 3: y = -1
4  Saída para a dígito 4: y = -1
5  Saída para a dígito 5: y = -1
6  Saída para a dígito 6: y = -1
7  Saída para a dígito 7: y = 1
8  Saída para a dígito 8: y = -1
9  Saída para a dígito 9: y = -1
10 Saída para a dígito 0: y = -1
11 Entrada:

```

```

12
13
14 #####
15 #####
16 ###          ###
17             ###
18             ###
19             ###
20             ###
21             ###
22             ###
23             ###
24             ###
25             ###
26             ###
27             ###
28             ###
29             ###

```

```

32 Saída estimada:

```

```

35 #####
36 #####

```

```

37      ###          ###
38          ###
39          ###
40          ###
41          ###
42          ###
43          ###
44          ###
45          ###
46          ###
47          ###
48          ###
49          ###
50          ###

```

## 3.2 Teste 2

```

1      x = [
2      -1, -1, -1, -1, -1, 1, -1, -1, 1, -1, -1, -1, -1, -1, -1, -1, -1,
3      -1, -1, -1, -1, 1, 1, 1, 1, 1, 1, 1, 1, 1, -1, -1, -1,
4      -1, -1, 1, 1, 1, 1, -1, -1, 1, -1, -1, -1, 1, 1, 1,
5      -1, 1, 1, 1, -1, 1, -1, -1, -1, 1, -1, -1, -1, 1, 1, 1,
6      1, 1, 1, 1, -1, 1, -1, -1, -1, 1, -1, -1, -1, 1, 1, 1, -1,
7      1, 1, 1, 1, -1, 1, -1, -1, -1, 1, -1, -1, -1, 1, 1, 1, -1,
8      1, 1, 1, 1, -1, 1, -1, -1, -1, 1, -1, -1, -1, 1, 1, 1, -1,
9      -1, 1, 1, 1, 1, 1, -1, -1, -1, 1, -1, -1, 1, 1, 1, -1, -1,
10     -1, -1, 1, 1, 1, 1, -1, -1, -1, 1, -1, 1, 1, 1, -1, -1, -1, -1,
11     -1, -1, -1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, -1, -1, -1, -1, -1,
12     -1, -1, -1, -1, 1, 1, 1, 1, 1, 1, 1, 1, -1, -1, -1, -1,

```

```

13  -1,  -1, -1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, -1, -1, -1,
    -1, -1, -1,
14  -1, -1, 1, 1, 1, 1, -1, -1, -1, 1, -1, -1, 1, 1, 1, -1, -1,
    -1, -1, -1,
15  -1, 1, 1, 1, -1, 1, -1, -1, -1, 1, -1, -1, -1, 1, 1, 1, -1,
    -1, -1, -1,
16  -1, 1, 1, 1, -1, 1, -1, -1, -1, 1, -1, -1, -1, 1, 1, 1, 1,
    -1, -1, -1,
17  -1, 1, 1, 1, -1, 1, -1, -1, -1, 1, -1, -1, -1, 1, 1, 1, 1,
    -1, -1, -1,
18  -1, 1, 1, 1, -1, 1, -1, -1, -1, 1, -1, -1, -1, 1, 1, 1, 1,
    -1, -1, -1,
19  -1, -1, 1, 1, 1, 1, -1, -1, 1, -1, -1, -1, -1, 1, 1, 1,
    1, -1, -1, -1, -1,
20  -1, -1, -1, -1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, -1, -1,
    -1, -1, -1, -1,
21  -1, -1, -1, -1, -1, 1, -1, -1, -1, 1, -1, -1, -1, -1, -1, -1,
    -1, -1, -1
22  ]

```

```

1  Saída para a dígito 1: y = -1
2  Saída para a dígito 2: y = -1
3  Saída para a dígito 3: y = -1
4  Saída para a dígito 4: y = -1
5  Saída para a dígito 5: y = -1
6  Saída para a dígito 6: y = -1
7  Saída para a dígito 7: y = -1
8  Saída para a dígito 8: y = 1
9  Saída para a dígito 9: y = -1
10 Saída para a dígito 0: y = -1
11 Entrada:
12      #  #          #
13      #####        #
14      #####  #    ###  #
15      ###  #    #    ###  #
16      #####  #    #    ###  #
17      #####  #    #    ###  ##
18      #####  #    #    ###  #
19      #####    #    ###  #
20      #####    #    ###  #
21      #####
22      #####

```

```

23      #####
24      #####  #  ###
25      ###  #  #  ###
26      ###  #  #  #####
27      ###  #  #  #####
28      ###  #  #  #####
29      #####  #  #####
30      #####
31      #  #
32  Saída estimada:
33
34      #####
35      ###          ###
36      ###          ###
37      #####          ###
38      #####          ###
39      #####          ###
40      #####          ###
41      #####          ###
42      #####
43      #####
44      #####
45      ###          ###
46      ###          ###
47      ###          #####
48      ###          #####
49      ###          #####
50      ###          #####
51      #####

```

### 3.3 Teste 3

```

1      x = [-1, -1, 1, -1, -1, -1, -1, -1, 1, -1, -1, -1, -1, -1, 1,
          -1, -1, -1, -1, -1, -1, 1, -1, -1, 1, 1, 1, -1, 1, -1, 1, 1,
          1, 1, 1, -1, -1, 1, -1, -1, -1, 1, 1, -1, -1, 1, 1, 1, 1,
          1, 1, 1, -1, 1, 1, -1, -1, -1, -1, -1, 1, 1, 1, 1, -1, -1,
          -1, -1, 1, -1, 1, 1, 1, 1, 1, 1, 1, -1, -1, 1, -1, 1, 1, -1,
          -1, -1, -1, -1, -1, -1, -1, 1,
2  1, -1, -1, 1, -1, -1, -1, -1, 1, 1, -1, 1, -1, -1, -1, 1, -1, 1,
          -1, 1, 1, -1, 1, -1, -1, 1, -1, -1, 1, -1, -1, -1, -1, -1,
          -1, 1, 1, -1, 1, 1, -1, 1, -1, -1, -1, 1, -1, 1, -1, -1, -1,

```

```

1, -1, -1, -1, 1, -1, 1, 1, -1, -1, -1, -1, -1, -1, -1, -1, -1,
1, -1, 1, -1, -1, -1, 1, -1, 1, 1, -1, 1, -1, -1, 1, -1, -1,
-1, -1, -1, -1, -1, -1, -1, -1, 1, -1, 1, 1, -1, 1, 1, -1, 1,
-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, 1, -1, -1, -1, -1,
-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, 1, -1, 1, 1, 1, 1,
-1, -1, 1, -1, -1, 1, -1, 1, -1, -1, -1, -1, 1, -1, 1, 1, 1,
1, -1, -1, -1, -1, -1, -1, -1, 1, -1, -1, -1, -1, -1, 1, 1, -1,
-1, 1, -1, -1, -1, -1, -1, -1, -1, 1, 1, -1, -1, 1, -1, 1, 1,
-1, -1, 1, 1, 1, -1, 1, 1, -1, -1, -1, 1, 1, -1, -1, -1, -1,
-1, -1, -1, 1, 1, 1, 1, -1, -1, -1, -1, -1, -1, -1, 1, 1,
1, 1, -1, 1, 1, 1, 1, -1, 1, 1, 1, -1, 1, 1, 1, 1, 1, 1,
1, -1, -1, 1, 1, 1, 1, 1, 1, 1, 1, -1, 1, 1, 1, -1, -1, 1, 1,
-1, -1, -1, -1, 1, -1, -1, 1, -1, 1, -1, -1,
3 1, -1, -1, -1, 1, -1, -1, -1, -1, -1, 1, -1, -1, 1, 1, -1, -1, 1,
-1, 1, -1, -1, -1, -1, -1, -1, -1, -1, -1, 1, -1]

```

```

1 Saída para a dígito 1: y = -1
2 Saída para a dígito 2: y = 1
3 Saída para a dígito 3: y = -1
4 Saída para a dígito 4: y = -1
5 Saída para a dígito 5: y = -1
6 Saída para a dígito 6: y = -1
7 Saída para a dígito 7: y = -1
8 Saída para a dígito 8: y = -1
9 Saída para a dígito 9: y = -1
10 Saída para a dígito 0: y = -1
11 Entrada:
12 # # #
13 # ### # ##### #
14 ## ##### ##
15 ##### # ##### #
16 ## ## #
17 ## # # # # #
18 # ## ## # #
19 # # # ##
20 # # # ## # #
21 # ## ## #
22 #
23 # ##### # # #
24 # ##### #
25 ## # ## #
26 ## ### ## ##

```

```

27      #####          #####
28      #####   ###   #####
29      #####   ###   ###
30      #   #   #   #   #   #
31      ##   #   #           #
32      Saída estimada:
33
34      #####
35      #####
36      ###          #####
37      ##           #####
38      ##           #####
39              #####
40              #####
41              #####
42              #####
43              #####
44              #####
45              #####
46              #####
47              #####
48      #####          ###
49      #####
50      #####

```

## 4 Referências

- Link do diretório com código feito: <https://github.com/AugustoSoaresPorto/amaqufu>
- Link do material de apoio: [aqui](#)