

Ler como um robô

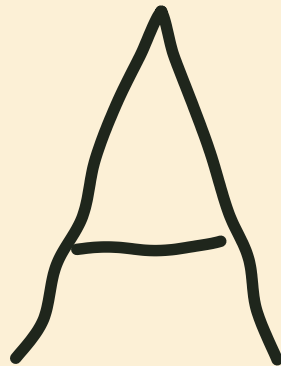
Como é que um computador reconhece letras escritas à mão?



Máquinas que leem

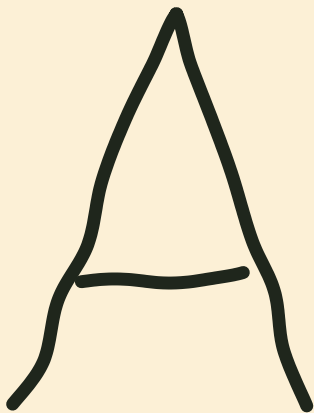
Há máquinas que leem o que escrevemos à mão:

- o telemóvel, quando escrevemos com o dedo no ecrã
- os correios, que leem o código postal nas cartas
- as portagens, que leem as matrículas dos carros
- os programas que passam documentos antigos a texto

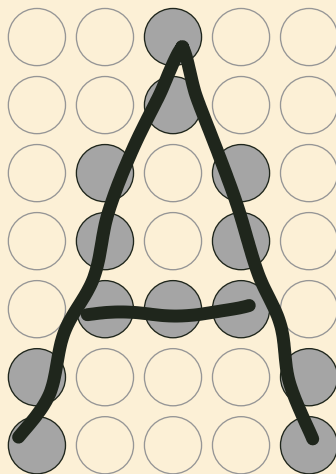


Como é que uma máquina, que só conhece 0 e 1, reconhece uma letra?

O que nós vemos e o que o robô vê



Nós vemos um A



Uma grelha de células



0	0	1	0	0
0	0	1	0	0
0	1	0	1	0
0	1	0	1	0
0	1	1	1	0
1	0	0	0	1
1	0	0	0	1

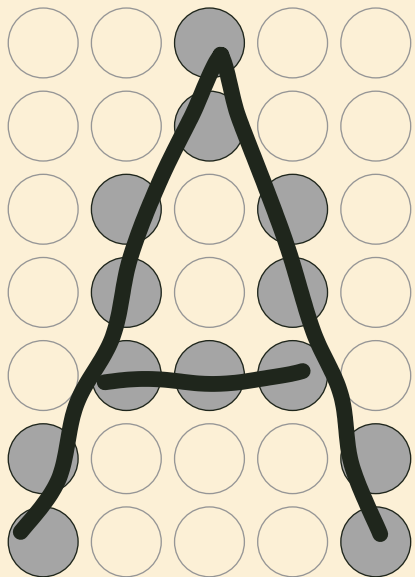
O robô vê 35 bits

Nível 1

Da letra aos bits

Jogador um: a imagem passa a 0 e 1.

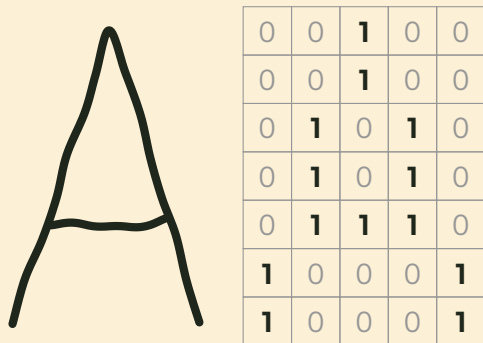
Passo 1: pôr uma grelha por cima da letra



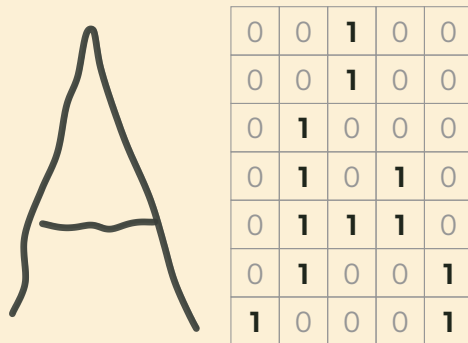
- Alguém escreveu este A à mão.
- Cada círculo é uma célula, um **píxel**.
- Se o traço passa pelo círculo, pintamos o círculo.
- Círculo pintado é **1**. Círculo vazio é **0**.
- 5 colunas × 7 linhas = **35 bits** por letra.

Chama-se binarizar: a imagem fica só com dois valores.

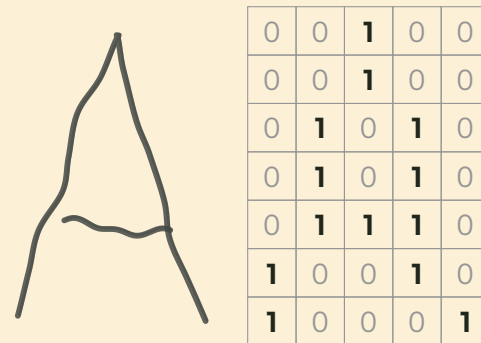
Três pessoas, três A



caneta azul, direita · 13 bits a 1



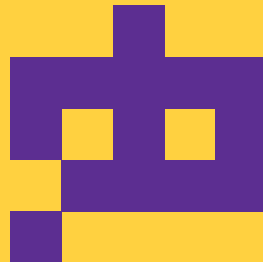
lápiz, inclinada · 12 bits a 1



lápiz fino, nervosa · 13 bits a 1

Ninguém escreve um A igual ao do vizinho: entre estes três há 3, 2 e 5 células diferentes. O robô vai ter de reconhecer todos.

Da letra aos bits



Cada um recebe uma letra escrita à mão por outra pessoa, com os círculos já pintados.

- Escreve **1** nas células pintadas e **0** nas vazias.
- Conta quantos bits estão a 1.
- Compara com um colega que tenha outra letra.

Nível 1. Carrega em START.

Nível 2

Da grelha ao vetor

As 7 linhas, uma a seguir à outra.

Passo 2: as linhas, uma a seguir à outra

0	0	1	0	0
0	0	1	0	0
0	1	0	1	0
0	1	0	1	0
0	1	1	1	0
1	0	0	0	1
1	0	0	0	1

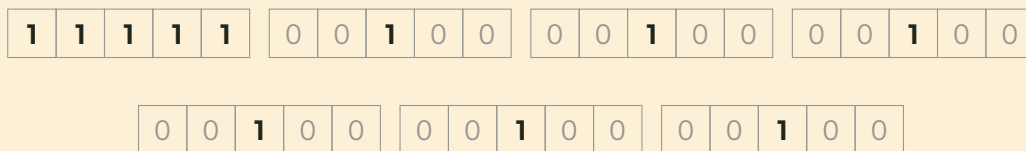
A memória de um computador é uma fila comprida de bits. Para guardar a grelha, escrevemos a linha 1, depois a linha 2, e assim até à linha 7.

0	0	1	0	0
0	0	1	0	0
0	1	0	1	0
0	1	0	1	0
0	1	1	1	0
1	0	0	0	1
1	0	0	0	1

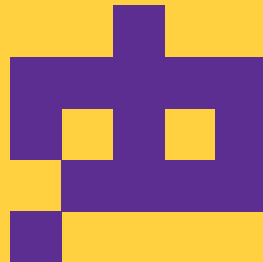
Uma fila de 35 bits: o **vetor** da letra. A isto chama-se aplanar a matriz.

Para que serve o vetor?

- Duas letras com o mesmo tamanho de grelha dão dois vetores com o mesmo comprimento: **35 bits**.
- Para comparar duas letras, basta comparar **posição a posição**: o bit 1 com o bit 1, o bit 2 com o bit 2, até ao bit 35.
- É assim que o computador guarda a letra, e é assim que a vai comparar com as letras que já conhece.



O vetor de um T: 35 bits.



Da grelha ao vetor

Agora os círculos vêm vazios.

- Pinta os círculos por onde a letra passa (lembra-te da regra).
- Escreve os bits na grelha e depois o vetor: as 7 linhas, uma a seguir à outra.
- Cartões 2.7 e 2.8: a letra és **tu** que a escreves, à mão, por cima dos círculos.

Nível 2. Sem rede.

Nível 3

Quem é a letra?

O robô compara com o que tem na memória.

A memória do robô

O robô guarda letras que já conhece, cada uma como uma grelha de 35 bits. Hoje a memória tem três letras:

À

0	0	1	0	0
0	0	1	0	0
0	1	0	1	0
0	1	0	1	0
0	1	1	1	0
1	0	0	0	1
1	0	0	0	1

O

0	1	1	1	0
1	0	0	0	1
1	0	0	0	1
1	0	0	0	1
1	0	0	0	1
1	0	0	0	1
0	1	1	1	0

↑

1	1	1	1	1
0	0	1	0	0
0	0	1	0	0
0	0	1	0	0
0	0	1	0	0
0	0	1	0	0
0	0	1	0	0

Comparar: 1 na mesma posição

Letra lida

A

O

↑

0	0	1	0	0
0	0	1	0	0
0	1	0	1	0
0	1	0	1	0
0	1	1	1	0
1	0	0	0	1
1	0	0	0	1

0	0	1	0	0	1
0	0	1	0	0	1
0	1	0	1	0	2
0	1	0	1	0	2
0	1	1	1	0	3
1	0	0	0	1	2
1	0	0	0	1	2

0	1	1	1	0	1
1	0	0	0	1	0
1	0	0	0	1	0
1	0	0	0	1	0
1	0	0	0	1	0
1	0	0	0	1	2
0	1	1	1	0	0

1	1	1	1	1	1
0	0	1	0	0	1
0	0	1	0	0	0
0	0	1	0	0	0
0	0	1	0	0	1
0	0	1	0	0	0
0	0	1	0	0	0

13

3

3

Linha a linha: em cada linha, contam-se as posições onde a letra lida e a letra da memória têm as duas um 1. No fim soma-se.

A mais parecida ganha

A

O

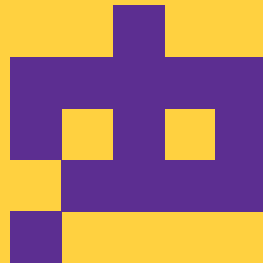
↑

13

3

3

O robô leu um **A**. Com apenas três letras na memória, acerta facilmente. E se tivesse 26?



Quem é a letra?

- Escreve os bits da letra lida.
- Para cada letra da memória, conta em cada linha os 1 que estão na mesma posição. Soma.
- A maior soma ganha: é a letra que o robô leu.

Nível 3. Três letras na memória.

Nível 4

O robô enganado

Uma medida que só conta o que coincide.

Um L contra L, E e F

Letra lida

L

E

F

1	0	0	0	0
1	0	0	0	0
1	0	0	0	0
1	0	0	0	0
1	0	0	0	0
1	0	0	0	0
1	0	0	0	0
1	1	1	1	1

1	0	0	0	0	1
1	0	0	0	0	1
1	0	0	0	0	1
1	0	0	0	0	1
1	0	0	0	0	1
1	0	0	0	0	1
1	0	0	0	0	1
1	1	1	1	1	5

11

1	1	1	1	1	1
1	0	0	0	0	1
1	0	0	0	0	1
1	0	0	0	0	1
1	1	1	1	0	1
1	0	0	0	0	1
1	0	0	0	0	1
1	1	1	1	1	5

11

1	1	1	1	1	1
1	0	0	0	0	1
1	0	0	0	0	1
1	1	1	1	0	1
1	0	0	0	0	1
1	0	0	0	0	1
1	0	0	0	0	1
1	0	0	0	0	1

7

Empate: o E contém todos os 1 do L, por isso coincide tanto como o L. Esta medida só vê o que coincide e ignora o que sobra.

Pior: a mancha

Letra lida

A

O



0	1	1	1	0
1	0	0	0	1
1	0	0	0	1
1	0	0	0	1
1	0	0	0	1
1	0	0	0	1
0	1	1	1	0

0	0	1	0	0	1
0	0	1	0	0	0
0	1	0	1	0	0
0	1	0	1	0	0
0	1	1	1	0	0
1	0	0	0	1	2
1	0	0	0	1	0

3

0	1	1	1	0	3
1	0	0	0	1	2
1	0	0	0	1	2
1	0	0	0	1	2
1	0	0	0	1	2
1	0	0	0	1	2
0	1	1	1	0	3

16

1	1	1	1	1	3
1	1	1	1	1	2
1	1	1	1	1	2
1	1	1	1	1	2
1	1	1	1	1	2
1	1	1	1	1	2
1	1	1	1	1	3

16

Se a memória tiver uma mancha toda a 1, a mancha empata com qualquer letra, ou ganha. Precisamos de uma medida melhor.

Contar diferenças

Letra lida

L

E

F

1	0	0	0	0
1	0	0	0	0
1	0	0	0	0
1	0	0	0	0
1	0	0	0	0
1	0	0	0	0
1	1	1	1	1

1	0	0	0	0	0
1	0	0	0	0	0
1	0	0	0	0	0
1	0	0	0	0	0
1	0	0	0	0	0
1	0	0	0	0	0
1	1	1	1	1	0

1	1	1	1	1	4
1	0	0	0	0	0
1	0	0	0	0	0
1	1	1	1	0	3
1	0	0	0	0	0
1	0	0	0	0	0
1	1	1	1	1	0

1	1	1	1	1	4
1	0	0	0	0	0
1	0	0	0	0	0
1	1	1	1	0	3
1	0	0	0	0	0
1	0	0	0	0	0
1	0	0	0	0	4

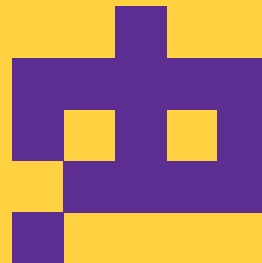
0

7

11

Agora contamos as posições **diferentes** (1 contra 0). Quanto menor a soma, mais parecida. O L ganha sozinho: 0 diferenças. Chama-se distância de Hamming.

O robô enganado



- Fichas 4.1 e 4.2: conta os 1 na mesma posição. Repara no que corre mal.
- Fichas 4.3 a 4.5: conta as **diferenças**. A menor soma ganha.

Nível 4. Uma medida nova.

Nível 3

Desafios

Para quem quer ver o robô a falhar.

Quatro desafios

§.1 A letra deslocada

A mesma letra, escrita mais à direita. O robô ainda a reconhece?

§.2 Letras parecidas

Que letras da memória estão mais perto uma da outra? Essas são as que o robô confunde.

§.3 Memória maior

Cinco letras na memória. Quem ganha, e quem fica em segundo?

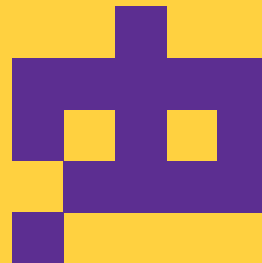
§.4 Inventa uma letra

Desenha uma letra e dá-a a um colega. Consegues enganar o robô?

Desafios

Em pares. Cada par escolhe por onde começa.

- Conta sempre **diferenças**: a menor soma ganha.
- Quando o robô se enganar, escreve porquê.



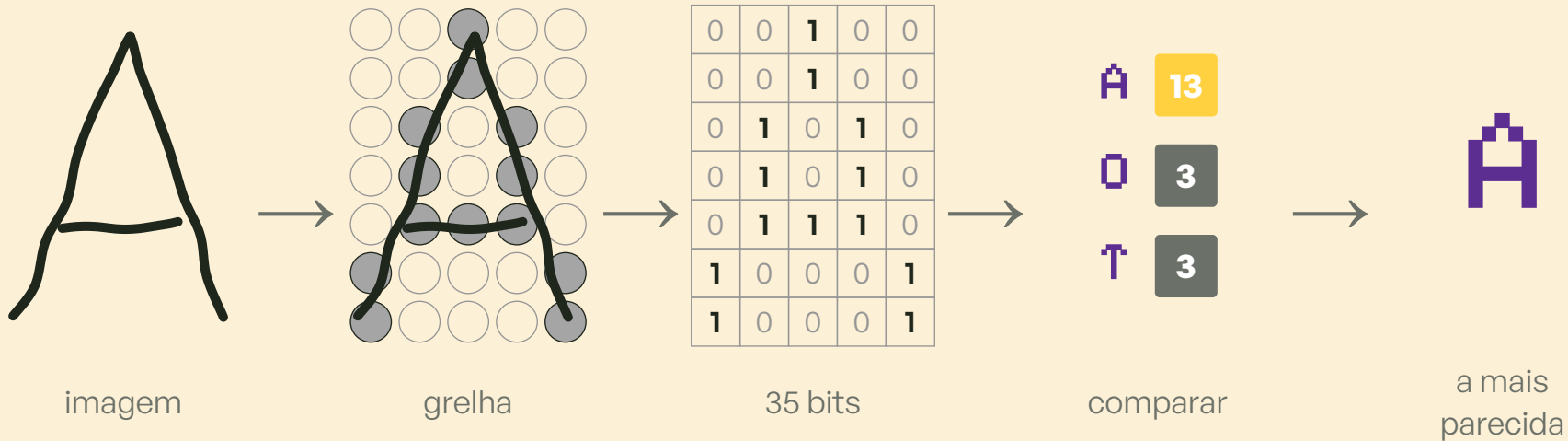
Nível S. Bónus.

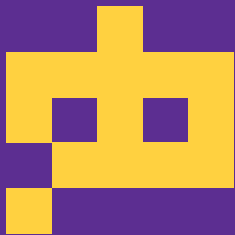
É os robôs a sério?

O nosso método é real: chama-se **comparar com o vizinho mais próximo**. Os sistemas a sério fazem o mesmo, mas:

- usam grelhas muito maiores (por exemplo $28 \times 28 = 784$ píxeis) e **tons de cinzento**, não só 0 e 1;
- guardam **milhares de exemplos** de cada letra, escritos por muitas pessoas, e não um só;
- **centram e ajustam** a letra antes de comparar, para a posição e o tamanho não contarem;
- as redes neuronais **aprendem** que traços distinguem as letras, em vez de comparar píxel a píxel;
- usam o **contexto**: depois de «Q» vem quase sempre «U».

O que o robô fez





Fim de nível

Primeiros Bits · CoTEDI · TreeTree2