

TEACHER'S GUIDE

READING LIKE A ROBOT

How does a computer recognise handwritten letters?

The pupils do by hand what a character recogniser does inside: they turn a handwritten letter into a grid of 0s and 1s, write it as a vector, compare it with the letters the robot has in its memory, and find out where the method fails and how to fix it.

Programme	Faíscas Years 4 to 6 in Portugal (ages 9–12); can be adapted up to Year 9 (age 15)
Pupils	
Duration	60 minutes (45-minute version)
Materials	Projector, printed A6 cards, pencils and rubbers
Organisation	Individually in levels 1 to 4; in pairs in level 5
Version	1.0



OVERVIEW

The question. A computer only stores 0s and 1s. So how does it recognise a letter we write by hand on a screen or on an envelope? The activity answers with a real method, simplified enough to fit on an A6 card: the picture becomes a grid of 35 bits, the grid becomes a vector, and the vector is compared with the vectors of the letters the robot already knows. The most similar one is the answer.

What the pupils learn. By the end of the lesson, the pupils should be able to:

- turn a simple picture into a grid of 0s and 1s (binarise it) and explain what is lost;
- write that grid as a single row of bits (a vector) and say what it is for;
- compare two vectors position by position and decide which is more similar, using a simple measure;
- explain why counting only the matches fools the robot, and why counting the differences (the Hamming distance) solves the problem;
- point out two or three ways in which real recognisers go further.

What the pupils do. Five levels of A6 cards, each with one new step: writing bits (level 1), deciding and writing the vector (level 2), recognising a letter by comparing it with the memory (level 3), finding the flaw in the measure and replacing it (level 4) and challenges in pairs (level 5). Each pupil gets a different letter in levels 1 and 2, which gives them something to compare with the classmate next to them.

Links to other lessons. In Faíscas, this lesson comes after «Portas lógicas» (Logic gates), where the pupils have already seen that a computer works with 0s and 1s, and before «Comandar um robô» (Commanding a robot). If the class has not done the previous lesson, the section «Background for the teacher» has what you need.

In brief

Duration	60 minutes; a 45-minute version without level 5
Organisation	individually in levels 1 to 4, in pairs in level 5; the whole class for the slides
Materials	projector; A6 cards (one per pupil and level, two in level 4); pencils and rubbers; a ruler, optional
Files	guia.pdf (this guide), slides.pdf (28 slides), notas.pdf (the slides with the notes for the teacher), fichas-a6.pdf (8 A4 pages, 4 cards per page), fichas-a4.pdf (one card per page, enlarged)

BEFORE THE LESSON

Prior knowledge

- Counting up to 35 and adding up seven small numbers.
- Reading a grid by rows and columns (like a table).
- Knowing what a bit is helps, but is not necessary: the activity defines it on slide 5 ("a coloured circle is 1, an empty one is 0").

What to prepare

- Open slides.pdf in full screen and check that the handwritten letter on slides 3 and 5 can be seen clearly from the back of the room.
- Print and cut out the cards (see the table). Cut each sheet in half vertically and horizontally along the marks; the marks are in the margins and in the centre of each sheet.

- Sort the cards by level into five piles. Shuffle each pile: pupils sitting next to each other should get different letters.
- A pencil and a rubber for each pupil. A ruler helps to follow the rows of the grid in levels 3 and 4, but is not necessary.

What to print

The cards are in `fichas-a6.pdf`, in black and white, four A6 cards per A4 sheet, each sheet with cards from one level only. We recommend printing single-sided, because the cards are cut out and handed out one by one.

Materials	Quantity	Format
Level 1 — From the letter to the bits (pages 1 and 2, cards 1.1 to 1.8)	1 card per pupil	class of 24: 3 copies of pages 1 and 2
Level 2 — From the grid to the vector (pages 3 and 4, cards 2.1 to 2.8)	1 card per pupil	class of 24: 3 copies of pages 3 and 4
Level 3 — Which letter is it? (pages 5 and 6, cards 3.1 to 3.8)	1 card per pupil	class of 24: 3 copies of pages 5 and 6
Level 4 — The robot fooled (page 7, cards 4.1 to 4.4)	2 cards per pupil: one of 4.1/4.2 and one of 4.3/4.4	class of 24: 12 copies of page 7
Level 5 — Challenges (page 8, cards 5.1 to 5.4); only in the 60-minute lesson	1 card per pair; the pairs swap	class of 24: 3 copies of page 8

General rule for a class of n pupils: levels 1, 2 and 3, $\frac{n}{8}$ copies of their two pages (rounded up); level 4, $\frac{n}{2}$ copies of page 7; level 5, $\frac{n}{8}$ copies of page 8. In the 60-minute lesson, 24 pupils need 33 A4 sheets and 28 pupils need 42. In the 45-minute version, without level 5, 24 pupils need 30 A4 sheets and 28 pupils need 38. The class pack, on the lesson page, does these sums for any class and puts the pages in the right order.

`fichas-a4.pdf` has the same 32 cards, one per page, enlarged to twice the size. It is for pupils who need larger print and cells, for projecting a card, and for anyone who prefers each exercise on its own page.

Timings

Stage	Minutes	What happens
Opening	5	Slides 1–3: machines that read; what we see and what the robot sees
Level 1 — From the letter to the bits	10	Slides 4–8; cards 1.1 to 1.8
Level 2 — From the grid to the vector	10	Slides 9–12; cards 2.1 to 2.8
Level 3 — Which letter is it?	12	Slides 13–17; cards 3.1 to 3.8
Level 4 — The robot fooled	10	Slides 18–22; cards 4.1 to 4.4
Level 5 — Challenges	8	Slides 23–25; cards 5.1 to 5.4, in pairs
Closing and checking	5	Slides 26–28; five spoken questions
Total	60	

45-minute version. Opening 4, level 1 8, level 2 8, level 3 12, level 4 8 (one card of 4.1/4.2 and one of 4.3/4.4, as in the 60-minute lesson), closing 5. Level 5 is left for those who finish early or for a second lesson.

LESSON PLAN

0 Opening 5 min slides 1–3

Show slide 2 and ask where they have seen a machine reading handwritten letters. Collect two or three examples (mobile phone, post office, number plates, scanning old documents) and ask the question of the day.

What to say

"A computer only stores 0s and 1s. So how does it recognise a handwritten A, which is never the same as the one written by the person next to you? Today we are going to do by hand what the computer does inside."

Slide 3: the same letter as seen by us, through a grid and by the robot. Do not explain the numbers yet; just show that the letter "becomes" 35 bits.

1 Level 1 — From the letter to the bits 10 min slides 4–8

Slide 5: an A handwritten by someone, with the grid of circles on top. Each circle is a cell (a pixel); if the stroke goes through the circle, the circle is coloured in; coloured in is 1, empty is 0. Do the conversion out loud for the first two rows, asking the pupils for the values.

What to say

"We read it like a book: from left to right, from top to bottom. First row: empty, empty, coloured, empty, empty. Which bits are they? 0 0 1 0 0."

Slide 6: the rule for circles the letter only just touches. Agree on a rule with the class (colour it in if the stroke goes through the middle; do not colour it in if it only grazes the edge) and say that two people may disagree on one or two cells; that is to be expected. Slide 7: the same A written by three people gives three different grids; this is the problem the robot has to solve, and that is why the cards have letters by different "writers" (upright pen, slanted pencil, thick pen, fine pencil).

Practice (slide 8). Give each pupil a level 1 card. The card shows the whole chain: the letter as it was handwritten, the grid with the circles the stroke goes through already coloured in (in grey, so the stroke can be seen), and the empty grid where the pupil writes the 35 bits and answers the two questions (how many bits set to 1 the letter has; how many bits the grid has: 35). Those who finish compare with their neighbour, who has a different letter.

Answers

All the grids are in the section «Solutions». How many bits the grid has: 35 (5 columns × 7 rows), on every card.

Watch out

Some pupils leave cells blank instead of writing 0. Insist: all 35 cells have a value.

2 Level 2 — From the grid to the vector ■ 10 min ■ slides 9–12

Slide 10: a computer's memory is a long row of bits. To store the grid, the rows are written one after another: row 1, then row 2, up to row 7. The result: a row of 35 bits, the letter's **vector**. Slide 11: what it is for. Two letters give two vectors of the same length, which can be compared position by position.

What to say

"Imagine a bookcase with 7 shelves of 5 books. To put them in a single line on the floor, you take the first shelf, then the second, and so on. The line is the vector; nothing gets lost, because we know that each group of five books comes from one shelf."

Practice (slide 12). Level 2 card: now the circles are empty. The pupil colours in the circles the letter goes through, writes the bits and then the vector, in the boxes underneath (rows 1 to 4 in the first line of boxes, 5 to 7 in the second). On cards 2.7 and 2.8 the pupils write the letter by hand themselves, over the circles (an E, and the first letter of their name): this is the moment when each pupil encodes their own handwriting. Give these two cards to pupils who finish the first one early, or to the whole class if there is time.

Answers

In the section «Solutions» (cards 2.1 to 2.6). Accept one or two cells that differ from the solution where the stroke passes close to the edge of the circle; what matters is that the rule was applied consistently. Cards 2.7 and 2.8 have no single solution: check that the letter fills the grid, that the rule was followed and that the vector copies the rows in the right order.

3 Level 3 — Which letter is it? ■ 12 min ■ slides 13–17

Slide 14: the robot has three letters in its memory (A, O and T), each as a grid of 35 bits. Slide 15: it reads a new letter and compares it with each letter in the memory, row by row, counting in each row the positions where both have a 1. The counts for each row are added up. Slide 16: the biggest total wins; the robot read an A.

What to say

"The robot does not know what an A is. It can only count. For each letter in its memory, it counts how many 1s are in the same place. The one with the most is the one it says it read."

Do the first row of each comparison with the class, on slide 15, before handing out the cards.

Practice (slide 17). Level 3 card. The pupil writes the bits of the letter read (handwritten by someone else; the circles are already coloured in), then, for each letter in the memory, counts in each row the 1s in the same position, writes them in the small boxes, adds them up in the large box and writes the winning letter.

Tip

Comparing grid with grid is the same as comparing the vectors position by position; it is only folded into 7 rows so that it is easier to follow with a finger or a ruler.

Answers

3.1 letter A: A 13, O 3, T 3 → A. **3.2** letter T: A 3, O 4, T 11 → T. **3.3** letter O: A 3, O 16, T 4 → O. **3.4** letter P: P 15, M 10, U 8 → P. **3.5** letter U: P 8, M 12, U 15 → U. **3.6** letter M: P 10, M 19, U 12 → M. **3.7** letter E: E 18, L 11, H 12 → E. **3.8** letter H: E 12, L 8, H 17 → H.

4**Level 4 — The robot fooled** ■ 10 min ■ slides 18–22

This level is for those who thought the method was too easy: the measure we used has a flaw, and the pupils are going to find it.

First hand out card 4.1 (an L against L, E and F) or 4.2 (an O against A, O and a blot that is all 1s), asking the pupils to count the matches as in level 3 and to look closely at the result. Let them reach the tie before you show the slides.

Answers

4.1 letter L, matches: L 11, E 11, F 7 → a tie, L / E. **4.2** letter O, matches: A 3, O 16, blot 16 → a tie, O / blot.

Slide 19: E contains all the 1s of L, so it matches as much as L itself; the measure only sees what matches and ignores what is left over. Slide 20: with a blot that is all 1s in the memory, the blot ties with any letter or wins, because it matches everything. Ask the class how to fix it; someone almost always suggests “also counting what is extra”.

Slide 21: counting differences. In each position, if one has a 1 and the other a 0, it is a difference. The smaller the total, the more alike the letter. L against L gives 0 differences; against E it gives 7; against F, 11. L wins on its own. Give it its name: the Hamming distance.

What to say

“Before, we only counted what matched. Now we count everything that does not fit: what is missing and what is extra. The right letter is the one with the fewest differences. The blot no longer wins, because all the extra ink it has counts as differences.”

Practice (slide 22). Cards 4.3 and 4.4, with the new measure. Card 4.3 has the same letter as 4.1, so the class can see the new measure breaking the tie.

Answers

4.3 letter L, differences: L 0, E 7, F 11 → L. **4.4** letter E, differences: L 7, E 0, F 4 → E. On card 4.2, with differences, the blot would give 19 and O would give 0: the blot no longer wins.

5**Level 5 — Challenges** ■ 8 min ■ slides 23–25

In pairs, with the differences measure. Each pair chooses where to start; they do not need to do all four. Slide 24 sums up the challenges.

- **5.1 The shifted letter.** The same I, written one column further to the right. Against I, T and L it gives I 14, T 14, L 16: a tie between I and T, and the right letter no longer wins on its own. Centred, it would give I 0, T 4, L 16. What to do before comparing: **centre** the letter in the grid (and, in real systems, also adjust its size).
- **5.2 Letters that look alike.** Differences between the models O, C, G and Q: O–C 3, O–G 3, O–Q 2, C–G 4, C–Q 5, G–Q 5. The most alike are O and Q (2 differences); the most different, C

and Q, and also G and Q (5). The conclusion to draw: the robot most easily confuses the letters whose vectors are closest; on such a small grid, O and Q can hardly be told apart.

- **5.3 A bigger memory.** An E against B, E, F, L and P: B 7, E 0, F 4, L 7, P 7. E wins; F comes second (4). With more letters in the memory, it is more likely that one of them is close to the right letter: the gap between first and second place may shrink, and the risk of a mistake grows.
- **5.4 Invent a letter.** No single solution. The pupils draw a letter (or a symbol), write the bits and swap with their classmate, who compares it with A, O, T, I and L. A letter the robot “reads wrongly” is one that is not in the memory or that is the same distance from two letters that are; this is exactly what happens in real systems when they meet something they have never seen.

6 Closing and checking 5 min slides 26–28

Slide 26: our method is real and has a name (comparing with the nearest neighbour); real systems do the same with bigger grids, shades of grey, thousands of examples per letter, centring first, neural networks that learn the important strokes, and context. Slide 27: the whole journey, from the picture to the letter. Finish with the questions in the section «Checking what they have learnt», out loud or on paper.

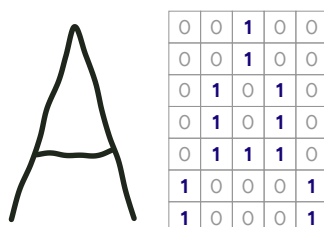
BACKGROUND FOR THE TEACHER

Binarising: from the picture to the bits

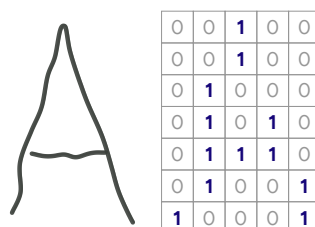
A camera captures millions of pixels, each with a colour. Here we reduce the picture to 35 cells (5 columns × 7 rows) and to two values: 1 if the letter goes through the cell, 0 if it does not. Information is lost, on purpose: that is the price of having something small that can be compared quickly. The 5×7 grid is the one used by many old text displays, where each character was drawn with 5×7 dots. Slide 6 deals with the decision for the cells on the boundary; in real systems the same difficulty leads to using shades of grey (how much of the pixel is covered) instead of just 0 and 1.

Everyone's handwriting

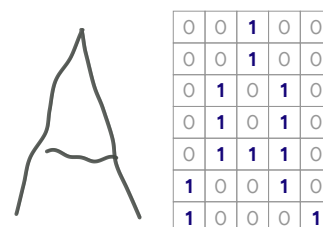
The letter that goes into the robot is always a letter written by someone, and no two people ever write the same letter in the same way: the slant, the width, a shaky hand and the pen all change the stroke, and the stroke changes the grid. The same A written by three people:



blue pen, upright · 13 bits set to 1 · 0 differences from the model



pencil, slanted · 12 bits set to 1 · 3 differences from the model



fine pencil, shaky · 13 bits set to 1 · 2 differences from the model

Between these three As there are 3, 2 and 5 different cells; the model A in the memory is 0, 3 and 2 differences away from each of them. That is why the activity always starts from handwriting (the cards have letters by four different hands, and on cards 2.7, 2.8 and 5.4 the hand is the pupil's own) and why a robot with only one model per letter ends up failing: the next section and level 5 show where.

Matrix and vector

The grid is a **matrix** of 7 rows and 5 columns. To store and compare it, it is flattened into a row of 35 bits, the **vector**: row 1, row 2, up to row 7. This is called flattening, and it is literally what happens in a computer's memory, where everything is a sequence. Two letters on the same grid give vectors of the same length, which makes it possible to compare position 1 with position 1, 2 with 2, up to 35. On the cards, the comparison is done row by row on the grids, which is the same thing without the row of 35 digits, which is harder to follow by eye.

Measuring similarity: matches or differences

With two vectors x and y of 35 bits, the measure in levels 3 and 4 (first part) counts the positions where **both** have a 1:

$$\text{matches}(x, y) = \sum_{i=1}^{35} x_i \cdot y_i$$

This is the **dot product** of the two vectors. The bigger it is, the more alike. The flaw: it only sees what matches. A letter in the memory that **contains** the letter read (E contains L) ties with the right answer, and a vector that is all 1s (the "blot") matches everything, so it always ties or wins. In other words, the measure favours whoever has more 1s.

The measure in level 4 (second part) counts the positions where the two vectors **differ**:

$$\text{differences}(x, y) = \sum_{i=1}^{35} |x_i - y_i|$$

This is the **Hamming distance**. The smaller it is, the more alike; the letter itself gives 0; the blot gives 35 minus the number of 1s in the letter read, because each 0 in the letter is a difference. That is why the blot no longer beats the right letter, although it may end up closer than another letter in the memory (on card 4.2, the O read is closer to the blot than to A). The worst possible result, 35, is that of the opposite vector, with a 1 wherever the letter has a 0 and a 0 wherever it has a 1. The Hamming distance counts what is missing and what is extra, and so it does not have the earlier flaw. For vectors of 0s and 1s it is also the square of the Euclidean distance, the "straight-line distance" between two points, which links this activity to the geometry the pupils know.

Ties

With either measure there can be a tie (card 5.1: the shifted I is the same distance from I and from T). A real system can choose at random, ask for more information (which letter comes before and after it in the word), or declare that it is not sure. A good question for the class: "If you were the robot, what would you do?"

What real recognisers do differently

The activity's method is the **nearest neighbour**: examples are stored and the most similar one is chosen. It is a real method and, with a bigger grid and many examples per letter, it works well for handwritten digits. Real systems differ in five ways that are worth telling the pupils, without exaggerating any of them:

1. **Bigger grids and shades of grey.** A classic reference, the MNIST set of handwritten digits, uses $28 \times 28 = 784$ pixels, each with a shade of grey, instead of 35 bits.
2. **Many examples per letter.** Instead of one model for each letter, thousands of examples written by different people; the letter read is compared with all of them, and the decision can be made by a majority of the k nearest.
3. **Normalisation.** Before comparing, the letter is centred and resized, which solves the problem of card 5.1. Systems are also trained with shifted and rotated versions of the examples.

4. **Learnt features.** A neural network does not compare pixel by pixel; from the examples, it learns which combinations of pixels (a horizontal stroke in the middle, a closed curve) tell the letters apart.
5. **Context.** A word reader uses the letters before and after, and a dictionary; a number-plate reader knows the format of number plates.

The table of differences on card 5.2 shows what stays the same: even in real systems, confusions happen between the classes whose representations are closest.

SOLUTIONS

Each handwritten letter was drawn afresh for this edition, by four “writers” with different hands; the solution is the grid obtained with the rule on slide 6. In levels 1 and 3 the circles are already coloured in, so there is no leeway; in level 2, accept one or two different cells near the edge.

Level 1 — From the letter to the bits

1.1 T, 11 bits set to 1

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
0	0	1	0	0

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

1.2 L, 11 bits set to 1

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

1.3 O, 16 bits set to 1

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	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	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

1.4 A, 13 bits set to 1

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

1.5 E, 18 bits set to 1

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

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

1.6 H, 16 bits set to 1

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

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

1.7 U, 15 bits set to 1

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

1.8 Z, 17 bits set to 1

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

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

Level 2 — From the grid to the vector

2.1 I, 11 bits set to 1

0	1	1	1	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	1	1	1	0

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

2.2 C, 13 bits set to 1

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

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

2.3 F, 14 bits set to 1

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

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

2.4 N, 20 bits set to 1

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

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

2.5 S, 15 bits set to 1

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

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

2.6 M, 19 bits set to 1

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

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

2.7 2.8 the pupils write the letter themselves; no single solution.

Levels 3 and 4 — comparisons

For each card: the grid of the letter read and, for each letter in the memory, the seven counts per row and the total. In level 3 and on cards 4.1 and 4.2, matches are counted (the biggest wins); on cards 4.3 and 4.4, differences are counted (the smallest wins).

3.1 letter A · matches

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

memory	r.1	r.2	r.3	r.4	r.5	r.6	r.7	total	
A	1	1	2	2	3	2	2	13	wins
O	1	0	0	0	0	2	0	3	
T	1	1	0	0	1	0	0	3	

3.2 letter T · matches

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

memory	r.1	r.2	r.3	r.4	r.5	r.6	r.7	total	
A	1	1	0	0	1	0	0	3	
O	3	0	0	0	0	0	1	4	
T	5	1	1	1	1	1	1	11	wins

3.3 letter O · matches

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

memory	r.1	r.2	r.3	r.4	r.5	r.6	r.7	total	
A	1	0	0	0	0	2	0	3	
O	3	2	2	2	2	2	3	16	wins
T	3	0	0	0	0	0	1	4	

3.4 letter P · matches

1	1	1	1	0	memory	r.1	r.2	r.3	r.4	r.5	r.6	r.7	total	
1	0	0	0	1	P	4	2	2	4	1	1	1	15	wins
1	0	0	0	1	M	1	2	2	2	1	1	1	10	
1	1	1	1	0	U	1	2	2	1	1	1	0	8	
1	0	0	0	0										
1	0	0	0	0										

3.5 letter U · matches

1	0	0	0	1	memory	r.1	r.2	r.3	r.4	r.5	r.6	r.7	total	
1	0	0	0	1	P	1	2	2	1	1	1	0	8	
1	0	0	0	1	M	2	2	2	2	2	2	0	12	
1	0	0	0	1	U	2	2	2	2	2	2	3	15	wins
0	1	1	1	0										

3.6 letter M · matches

1	0	0	0	1	memory	r.1	r.2	r.3	r.4	r.5	r.6	r.7	total	
1	1	0	1	1	P	1	2	2	2	1	1	1	10	
1	1	0	1	1	M	2	4	4	3	2	2	2	19	wins
1	0	1	0	1	U	2	2	2	2	2	2	0	12	
1	0	0	0	1										
1	0	0	0	1										

3.7 letter E · matches

1	1	1	1	1	memory	r.1	r.2	r.3	r.4	r.5	r.6	r.7	total	
1	0	0	0	0	E	5	1	1	4	1	1	5	18	wins
1	0	0	0	0	L	1	1	1	1	1	1	5	11	
1	1	1	1	0	H	2	1	1	4	1	1	2	12	
1	0	0	0	0										
1	0	0	0	0										

3.8 letter H · matches

1	0	0	0	1	memory	r.1	r.2	r.3	r.4	r.5	r.6	r.7	total	
1	0	0	0	1	E	2	1	1	4	1	1	2	12	
1	1	1	1	1	L	1	1	1	1	1	1	2	8	
1	0	0	0	1	H	2	2	2	5	2	2	2	17	wins
1	0	0	0	1										
1	0	0	0	1										

4.1 letter L · matches

1	0	0	0	0	memory	r.1	r.2	r.3	r.4	r.5	r.6	r.7	total	
1	0	0	0	0	L	1	1	1	1	1	1	5	11	tie
1	0	0	0	0	E	1	1	1	1	1	1	5	11	tie
1	0	0	0	0	F	1	1	1	1	1	1	1	7	
1	0	0	0	0										
1	1	1	1	1										

4.2 letter O · matches

0	1	1	1	0	memory	r.1	r.2	r.3	r.4	r.5	r.6	r.7	total	
1	0	0	0	1	A	1	0	0	0	0	2	0	3	
1	0	0	0	1	O	3	2	2	2	2	2	3	16	tie
1	0	0	0	1	blot	3	2	2	2	2	2	3	16	tie
1	0	0	0	1										
0	1	1	1	0										

4.3 letter L · differences

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

memory	r.1	r.2	r.3	r.4	r.5	r.6	r.7	total	
L	0	0	0	0	0	0	0	0	wins
E	4	0	0	3	0	0	0	7	
F	4	0	0	3	0	0	4	11	

4.4 letter E · differences

[illegible]

Level 5 — Challenges

5.1 letter l · differences

[illegible]

5.2 differences between the models:

	O	C	G	Q
O	0	3	3	2
C	3	0	4	5
G	3	4	0	5
Q	2	5	5	0

Most alike: O and Q; most different: C and Q, and also G and Q.

5.3 letter E · differences

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

memory	r.1	r.2	r.3	r.4	r.5	r.6	r.7	total	
B	1	1	2	0	1	1	1	7	
E	0	0	0	0	0	0	0	0	wins
F	0	0	0	0	0	0	4	4	
L	4	0	0	3	0	0	0	7	
P	1	1	1	0	0	0	4	7	

5.4 no single solution; see step 5.

COMMON MISTAKES AND HOW TO RESPOND

What happens	What to do
Mixing up 0 and 1, or changing the rule halfway through.	Repeat the reminder "1 is on, coloured in". Ask the pupil to say the rule out loud before carrying on.

What happens	What to do
Leaving cells blank instead of writing 0.	Remind them that the grid always has 35 values; a cell with nothing in it is not a 0, it is a reading error for the robot.
Swapping rows and columns in the vector.	"We read it like a book: from left to right, from top to bottom." Show the bookcase with shelves from step 2.
Colouring in, or not, the cells where the stroke only touches the edge.	Accept both options if the rule was followed consistently; it is the same doubt that real systems have, and it is the subject of slide 6.
Wrong counts per row in level 3.	Follow each row with a finger or a ruler on both grids; check in pairs; check the total: it can never be bigger than the number of 1s in the letter read.
In level 4, not seeing any problem with the tie.	Ask: "So did the robot read L or E? Can it answer both?" and "What would a blot that is all coloured in get?"
Mixing up the two measures on cards 4.3 and 4.4.	Read the title of the card: "Counting differences". In a difference, one has a 1 and the other a 0; positions with two 1s or two 0s do not count.
"Is this artificial intelligence?"	It is one of the simplest methods of pattern recognition, and a direct ancestor of today's: the principles (turn into numbers, compare, choose the most similar) are the same.
"What about letters that look alike, like O and Q?"	That is card 5.2: on such a small grid, O and Q differ in 2 cells. More cells or other features would be needed.

INCLUSION

The activity is all on paper, with no long reading, and each level has only one step; even so, some adaptations make a big difference for some pupils.

Reading difficulties

The cards have one line of instructions, but the task is explained out loud and by demonstration (slides 5, 10, 15 and 21): read the instruction out loud when handing out the card. Use fichas-a4.pdf (print at twice the size). Skip writing the vector as a row (level 2) if it is an obstacle: the bit grid already holds the same information.

Attention and organisation

Hand out one card at a time and collect the previous one. In levels 3 and 4, work on one letter in the memory at a time, covering the others with a sheet of paper. A ruler or a finger following the row reduces skipped rows. The levels are designed to last 8 to 12 minutes each; say how long they have.

Fine motor skills

Use fichas-a4.pdf, where the cells are 10 to 16 mm. Accept a cross or a line instead of the digit 1 and a blank cell as 0, as long as the pupil says the rule. In pairs, one pupil reads the bits out loud and the other writes them.

Vision

The cards are in black and white with high contrast and no colour that carries meaning. Project the card from `fichas-a4.pdf` or enlarge it further on the photocopier. Describe the handwritten letter out loud ("an A with the bar in the middle"). On the slides, all the information is also in the text, never only in the colour: the counts are written out.

Pupils who are more advanced or younger

More advanced: start level 4 without a demonstration, and move on to challenges 5.2 and 5.4. Younger (Year 4 in Portugal, ages 9–10): do levels 1 and 2 individually, level 3 together on the projector with only one letter in the memory, and leave level 4 for a second session.

EXTENSIONS

- **A whole alphabet.** Each pupil draws a different letter and writes its vector; the class puts the 26 together and, using the table of differences from card 5.2, looks for the pairs that are most alike.
- **Shades of grey.** Repeat level 1 with three values per cell (0, 1 and 2, depending on whether the letter covers little, some or a lot of the circle) and see whether the comparison becomes more reliable.
- **Several samples per letter.** Store three different As in the memory instead of one and decide by a majority among the closest; discuss why real systems store thousands.
- **Words and context.** Give the pupils a word in which one letter ties between two possibilities (for example, "QU_CK") and see how the context solves what the distance cannot.
- **Pictures instead of letters.** The next activity, «Ver como um robô» (Seeing like a robot), describes pictures by their features and uses the same kind of comparison.

CHECKING WHAT THEY HAVE LEARNT

Five quick questions, out loud or on half a sheet of paper, at the end of the lesson.

1. **How many bits does a grid of 5 columns and 7 rows have?** 35.
2. **What is the vector of a letter?** The rows of the grid written one after another, in a single row of 35 bits.
3. **Why does counting only the 1s that match fool the robot?** Because it ignores what is extra: a letter that contains the letter read, or a blot that is all 1s, ties or wins.
4. **How do you count the Hamming distance between two letters?** Position by position, counting the positions where one has a 1 and the other has a 0; the smaller it is, the more alike they are.
5. **Name one thing that real recognisers do and that we did not do.** Bigger grids with greys; many examples per letter; centring the letter before comparing; learning the important strokes; using the context. Any one of these will do.

SOURCES AND CREDITS

Activity created for Primeiros Bits (Instituto Superior Técnico and TreeTree2) and also used in the Faíscas programme. This edition reorganises the levels, adds levels 4 and 5, replaces the handwritten

letters with new drawings and brings together the background and the solutions. The letters, the grids and the solutions were generated by a program from the same skeleton for each letter, so the solutions can be checked. Contact: faiscas@treetree2.org · faiscas.pt

Conditions for using the materials

© Associação TreeTree2. Material of the Faíscas project.

Teachers with an approved account may use these materials with their own classes, share them with colleagues at the same school and adapt them for their lessons.

They may not be redistributed publicly (for example, posted on a website, on social media or on a platform for sharing teaching materials), nor sold. For other uses, write to faiscas@treetree2.org.