Nearest neighbor classification

Topics we'll cover

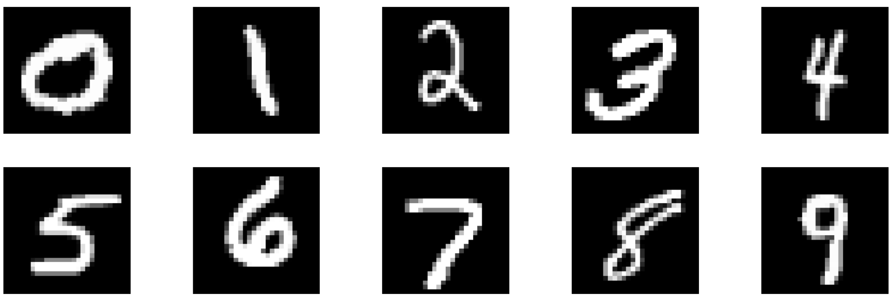
1. What is a classification problem?
2. The training set and test set
3. Representing data as vectors
4. Distance in Euclidean space
5. The 1-NN classifier
6. Training error versus test error
7. The error of a random classifier

The problem we'll solve today

Given an image of a handwritten digit, say which digit it is.

$\Longrightarrow$ 3

Some more examples:



The machine learning approach

Assemble a data set:

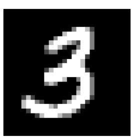
The MNIST data set of handwritten digits:

- **Training set** of 60,000 images and their labels.
- **Test set** of 10,000 images and their labels.

And let the machine figure out the underlying patterns.

Nearest neighbor classification

Training images $x^{(1)}, x^{(2)}, x^{(3)}, \dots, x^{(60000)}$
Labels $y^{(1)}, y^{(2)}, y^{(3)}, \dots, y^{(60000)}$ are numbers in the range $0-9$

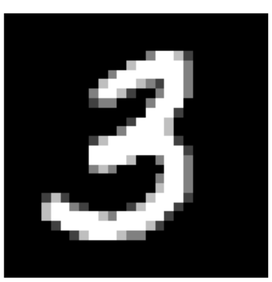


How to **classify** a new image x?

- Find its nearest neighbor amongst the $x^{(i)}$
- Return $y^{(i)}$

The data space

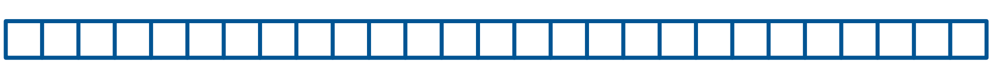
How to measure the distance between images?



MNIST images:

- Size 28×28 (total: 784 pixels)
- Each pixel is grayscale: 0-255

Stretch each image into a vector with 784 coordinates:

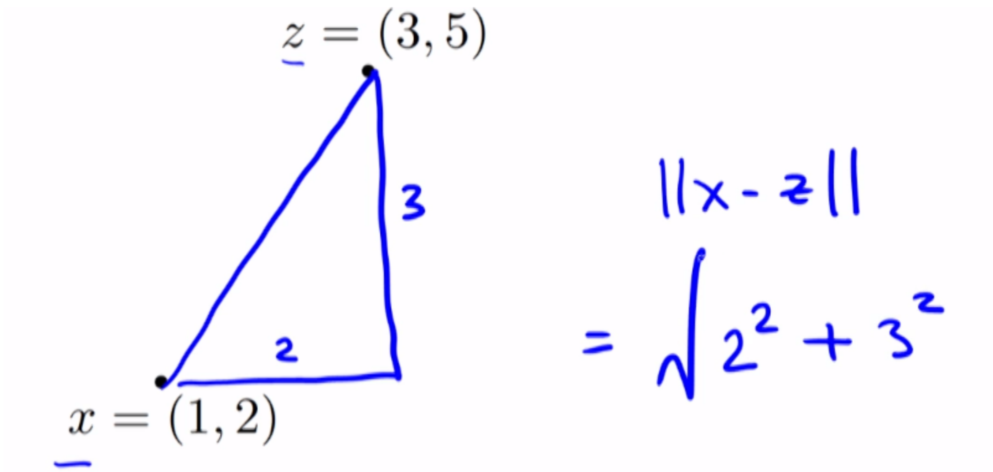


- Data space $\mathcal{X} = \mathbb{R}^{784}$
- Label space $\mathcal{Y} = \{0, 1, \dots, 9\}$

a 784 dimensional vector
Our data space then, which we're gonna denote by script X
is 784 dimensional Euclidean space R to the 784th.

The distance function

Remember Euclidean distance in two dimensions?



common, or default distance

function is perhaps just Euclidean distance.

When you have two points, the Euclidean distance

between them is just the length of the line connecting them.

So it's the length of this line.

And what is that length?

Well, if you look at these two points, X and Z,

along the first coordinate, they defer by two

and along the second coordinate, they defer by three.

So the length of the line, the distance from X to Z

is simply the square root of two squared plus three squared

which is the square root of 13.

That's the Euclidean distance between X and Z

Euclidean distance in higher dimension

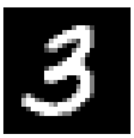
Euclidean distance between 784-dimensional vectors x, z is

$$\|x - z\| = \sqrt{\sum_{i=1}^{784} (x_i - z_i)^2}$$

Here x_i is the ith coordinate of x.

Nearest neighbor classification

Training images $x^{(1)}, \ldots, x^{(60000)}$, labels $y^{(1)}, \ldots, y^{(60000)}$



To classify a new image x:

- Find its nearest neighbor amongst the $x^{(i)}$ **using Euclidean distance in $\mathbb{R}^{784}$**
- Return $y^{(i)}$

How accurate is this classifier?

So we have these 60,000 training images.

For any training point, its nearest neighbor

in the training set is itself.

So it'll definitely get the right label.

So the error rate on the training set is zero.

What that means is that training error

is not a good predictor of future performance.

Accuracy of nearest neighbor on MNIST

Training set of 60,000 points.

- What is the error rate on training points? **Zero.**
  In general, **training error** is an overly optimistic predictor of future performance.

- A better gauge: separate test set of 10,000 points.
  Test error = fraction of test points incorrectly classified.

- What test error would we expect for a *random classifier*?
  (One that picks a label $0 - 9$ at random?) **90%.**

- Test error of nearest neighbor: **3.09%.**

Examples of errors

Test set of 10,000 points:

- 309 are misclassified
- Error rate 3.09%

Examples of errors: