

# Recognizing Individual Digits

---

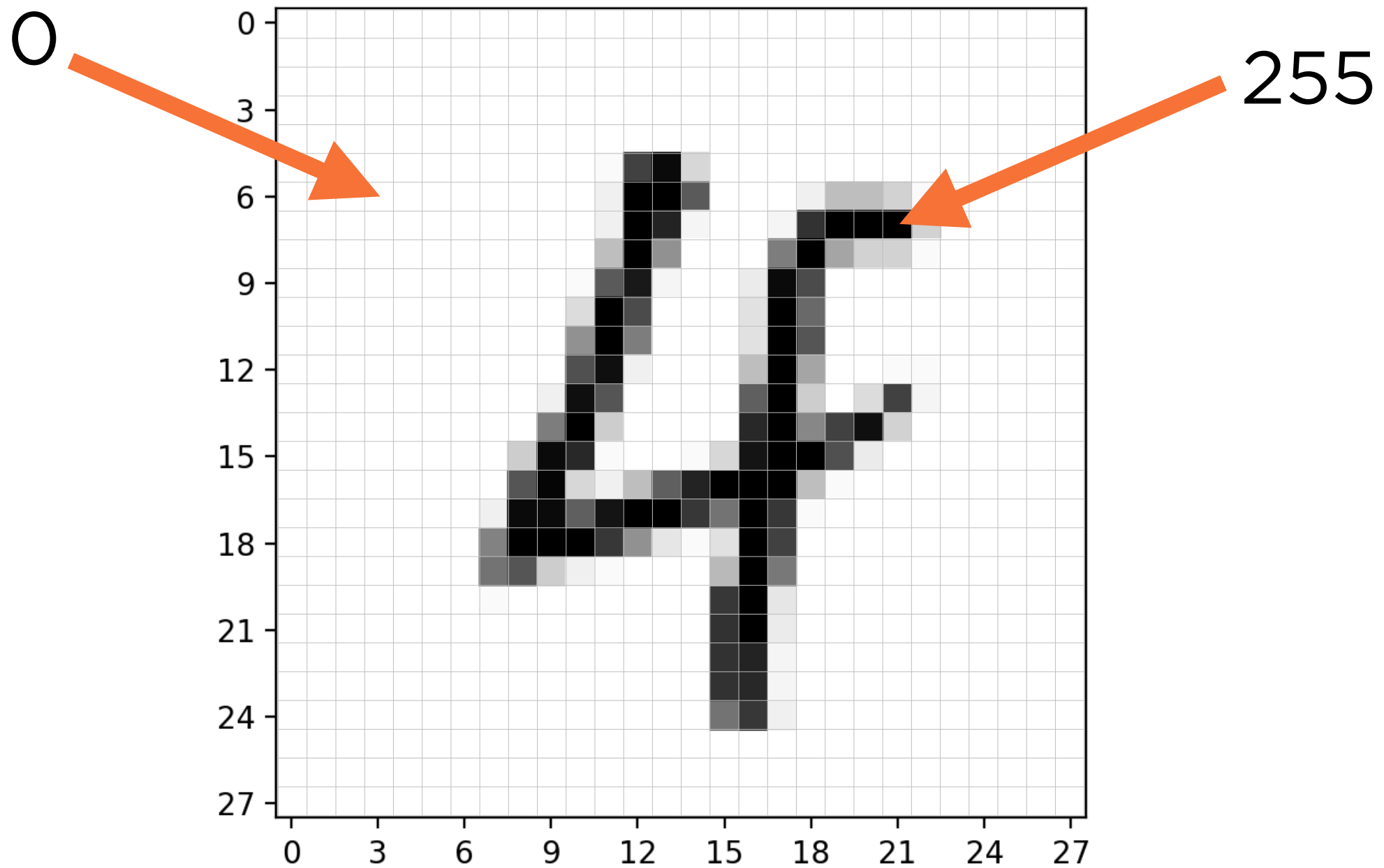


**Paolo Perrotta**

FREELANCE DEVELOPER

@nusco

# MNIST Digits



Label: 4

# Understanding the Test Set

---

# A Fake Classifier

```
DICTIONARY = {}

def predict(X):
    result = []
    for image in X:
        result << DICTIONARY[image]
    return result

def train(X, Y):
    for image in X:
        DICTIONARY[image] = Y

train(MNIST_IMAGES, MNIST_LABELS)
predictions = predict(MNIST_IMAGES)
```



# The Problem of Overfitting

## Without Overfitting

The system *generalizes* the training data. When confronted with new data, it can still classify it.

## With Overfitting

The system *memorizes* the training data. When confronted with new data, it doesn't know what to do.

Never test a system on the same  
data that you used to train it.

# Preparing MNIST Data for the Classifier

---

# MNIST Images vs. the X Matrix

8 9 8 2 9 4 9 5 7 6 3 3 6 3 8 2 0 7 9 4 2 0 8 4 2 1 0 5 6 7 2 0 3 9 1 0 0 3 0  
5 4 8 3 0 0 9 2 7 3 0 2 2 9 3 8 6 9 2 5 1 8 4 1 9 6 0 5 5 4 0 9 1 2 3 2 8 5 1 0  
5 7 6 6 3 6 2 4 1 6 1 2 0 7 2 3 0 7 1 4 7 3 7 7 0 7 4 7 5 5 1 8 2 5 4 0 8 9 2 6  
4 7 1 7 9 2 5 7 0 7 1 2 2 0 1 8 9 2 4 2 5 2 2 3 4 5 0 1 7 0 3 5 4 3 5 6 0 3 9  
8 6 6 2 9 1 8 6 0 6 9 8 4 0 6 8 3 0 5 1 4 1 5 2 6 5 5 1 9 7 0 1 3 4 7 4 8 2 3 8  
7 8 1 4 8 5 3 7 6 1 2 8 4 2 7 4 8 6 1 3 2 7 6 6 0 2 0 2 9 9 6 1 0 8 6 0 6 7 6 8  
6 3 8 1 9 2 4 5 3 0 9 3 6 6 1 0 8 0 7 3 5 0 7 9 7 9 8 5 7 6 5 5 4 3 7 9 3 2 3 9  
4 2 9 2 1 0 7 4 2 5 7 4 2 4 4 0 4 4 5 8 0 3 1 3 5 1 2 2 7 8 7 3 1 5 4 7 1 0 5 0  
0 1 3 7 7 0 9 8 4 5 9 1 6 0 0 9 0 7 8 4 0 6 6 3 7 6 3 9 0 4 3 2 4 0 6 8 5 0 7 0  
1 8 5 0 3 6 0 9 4 0 4 3 3 0 2 4 9 8 1 4 1 7 9 8 3 5 3 4 0 8 7 3 6 9 3 0 3 1 6 7  
4 6 0 9 4 0 8 8 1 0 5 2 8 8 5 8 5 3 7 4 0 9 9 9 6 8 8 8 3 0 6 1 6 1 9 6 0 9 6 4  
5 1 4 3 7 0 6 3 4 8 8 6 6 9 1 2 5 0 4 6 8 1 0 7 3 4 6 8 3 2 0 9 9 7 9 7 5 4 7 5  
6 0 9 1 6 2 2 2 1 1 6 9 4 5 1 7 6 2 3 2 5 1 4 6 3 8 7 4 4 2 1 1 6 5 1 2 3 1 3 5  
6 1 7 6 7 5 6 2 4 6 1 6 8 9 8 7 3 7 4 7 0 4 0 8 9 0 1 5 9 4 7 8 7 6 4 1 6 8 1 7  
3 5 5 6 8 7 4 5 5 3 6 7 0 5 1 0 8 9 9 7 2 8 5 1 6 0 1 0 1 2 7 7 1 3 7 9 2 3 2 5  
2 1 7 7 5 8 9 0 3 9 5 5 3 1 8 7 8 7 5 5 3 2 1 4 7 6 4 2 7 8 8 4 8 1 8 7 8 6 3 4  
2 4 5 7 4 2 8 5 8 9 8 0 6 0 1 7 9 0 7 3 7 1 3 0 7 4 8 5 3 3 0 1 7 6 5 7 0 0 5 8  
1 5 2 6 9 4 2 8 5 0 9 8 6 0 3 4 2 6 5 9 6 3 0 9 2 7 1 1 9 7 3 7 0 1 7 7 4 1 2 9  
5 8 9 8 7 9 5 7 2 7 9 4 2 8 2 1 7 8 0 8 1 5 4 4 9 5 4 9 9 3 3 8 4 7 4 7 7 5 3 1  
9 8 1 2 3 4 7 6 8 8 2 5 7 0 7 2 0 7 0 2 9 6 6 6 8 9 4 0 0 5 3 4 6 2 2 2 2 9 6 0



**X**

X <sub>0</sub>	X <sub>1</sub>	X <sub>2</sub>	...
X <sub>0</sub>	X <sub>1</sub>	X <sub>2</sub>	...
...	...	...	...

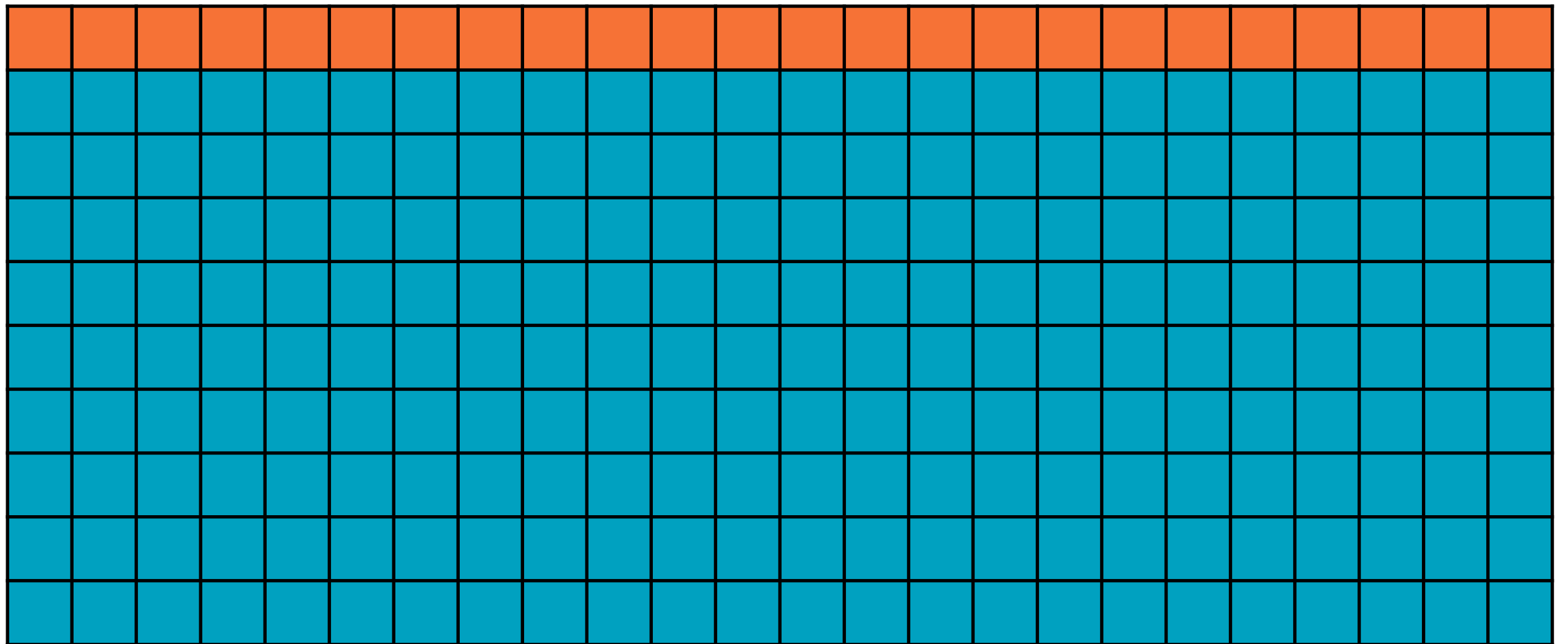


# MNIST Images vs. the X Matrix

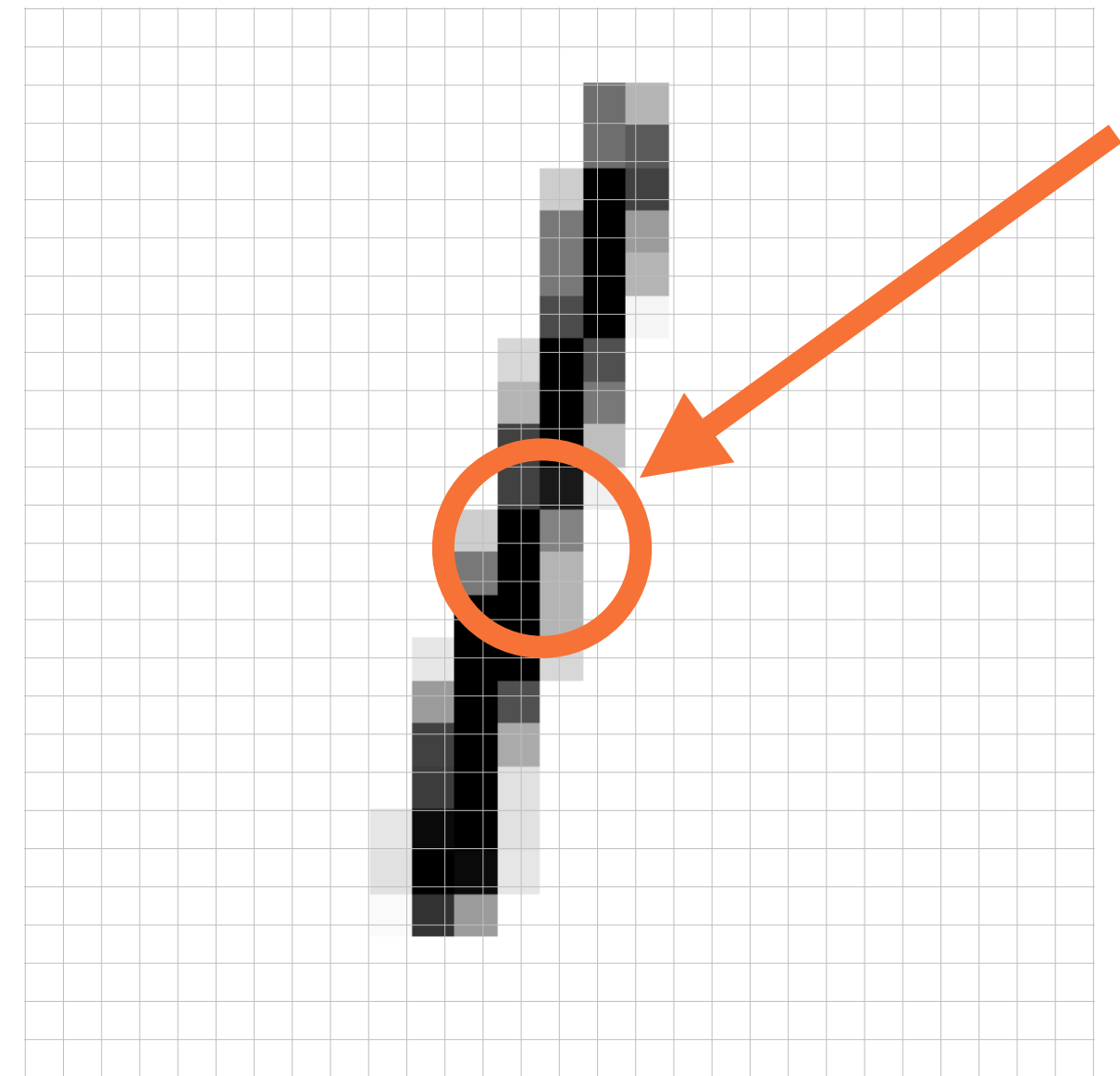
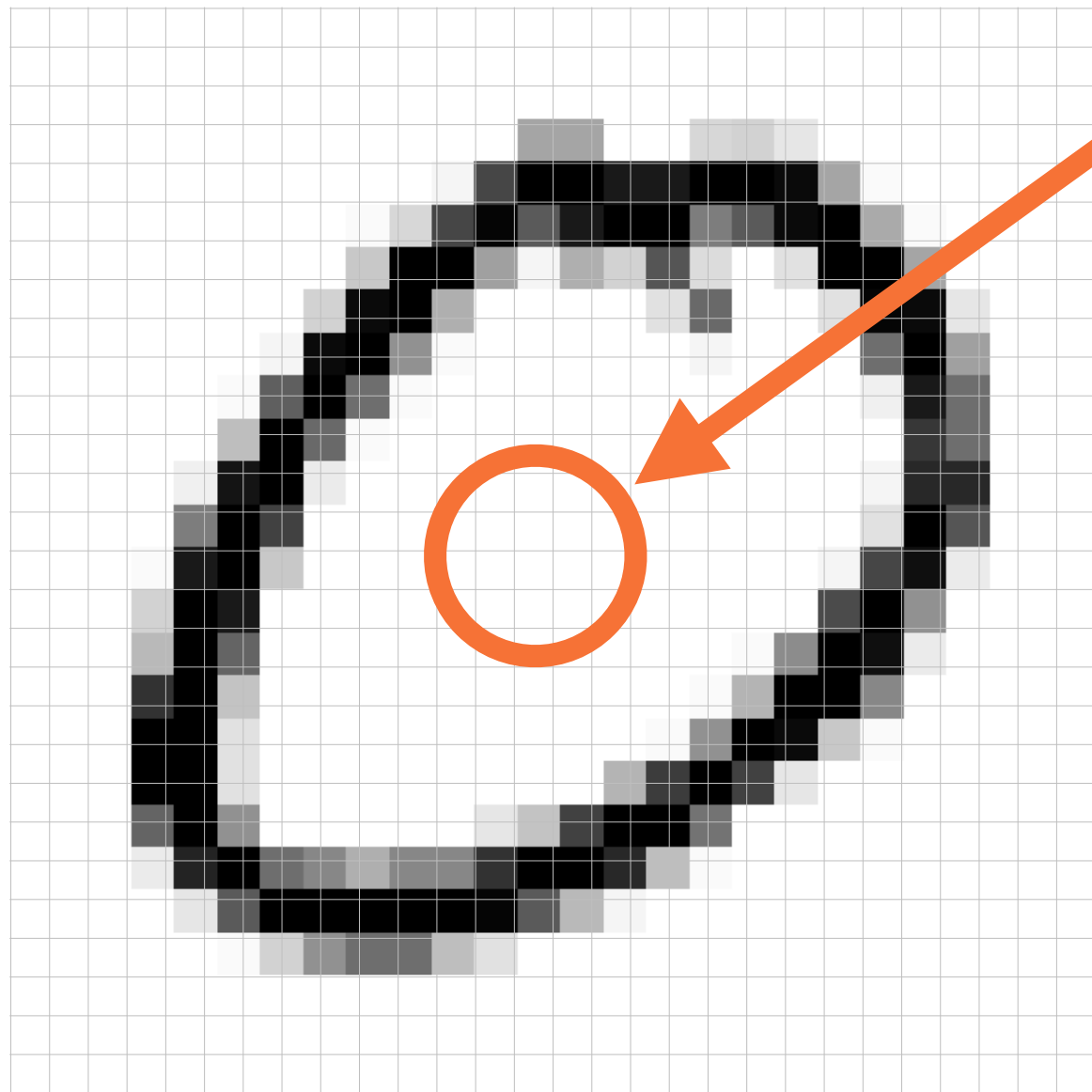
785  
columns

**X**

60K rows



# Recognizing Images from Pixel Values



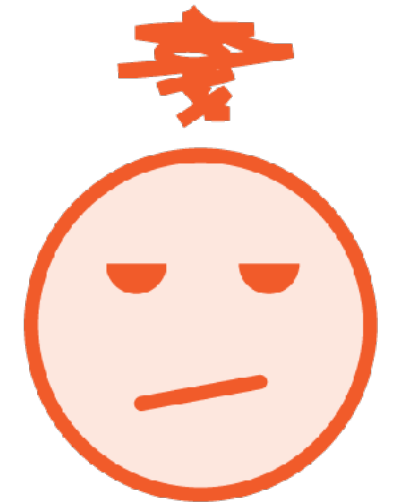
# MNIST Labels vs. the Y Matrix

4 3 9 1 7 8 4 9 4 2 6 8  
6 7 9 4 0 9 5 7 9 7 0 3  
9 3 1 2 3 0 5 8 4 9 7 ...



**Y**

4  
3  
9  
1  
7  
8  
4  
9  
4  
2  
6  
...



# MNIST Labels vs. the Y Matrix

4	3	9	1	7	8	4	9	4	2	6	8
6	7	9	4	0	9	5	7	9	7	0	3
9	3	1	2	3	0	5	8	4	9	7	...



**Y**

1
0
0
0
0
0
1
0
1
0
0
...

# Summary

**We introduced the MNIST dataset**

**We learned what the test set is all about**

**We encoded MNIST for our classifier**

- We flattened the images
- We encoded the “4” labels as 1s and the rest as 0s

**We ran the binary classifier on MNIST**