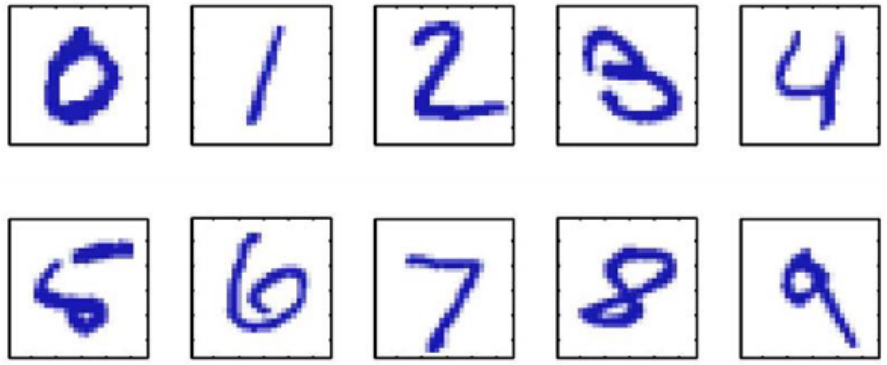
Example 1: Digit Recognition

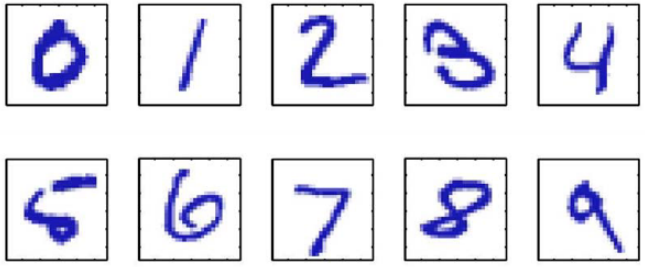


Images are 28 x 28 pixels

- Problem: Recognize a digit from the image
- MNIST dataset challenge
  - Dataset developed in 1990s to spur AI research on a challenging problem for the time
  - Data taken from census forms
  - Became a classic benchmark for machine vision problems
  - We will see this dataset extensively in this class

Classical “Expert” Approach

- Idea: Use your knowledge about digits
  - You are an “expert” since you can do the task
- Construct simple rules and code them
- Expert rule example: *“Image is a digit 7 if…”*:
  - There is a single horizontal line, and
  - There is a single vertical line
- Rule seems simple and reasonable
- But,…



Images are 28 x 28 pixels

```
def count_vert_lines(image):
    ...
def count_horiz_lines(image):
    ...

def classify(image):
    ...
    nv = count_vert_lines(image)
    nh = count_horiz_lines(image)
    ...

    if (nv == 1) and (nh == 1):
        digit = 7
    ...

    return digit
```

Problems with Expert Rules

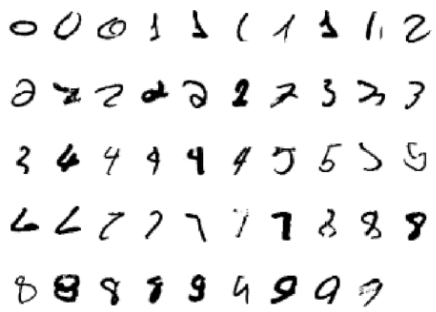
- Simple expert rule breaks down in practice
  - Hard to define a “line” precisely
  - Orientation, length, thickness, …
  - May be multiple lines…

- General problem: We cannot easily code our knowledge
  - We can do the task
  - But it is hard to translate to simple mathematical formula

```
def count_vert_lines(image):
    ...
def count_horiz_lines(image):
    ...

def classify(image):
    ...
    nv = count_vert_lines(image)
    nh = count_horiz_lines(image)
    ...

    if (nv == 1) and (nh == 1):
        digit = 7
    ...

    return digit
```

ML Approach: Learn from Data

Training inputs images $\boldsymbol{x}_i$ (ex. 5000 ex per class)



?

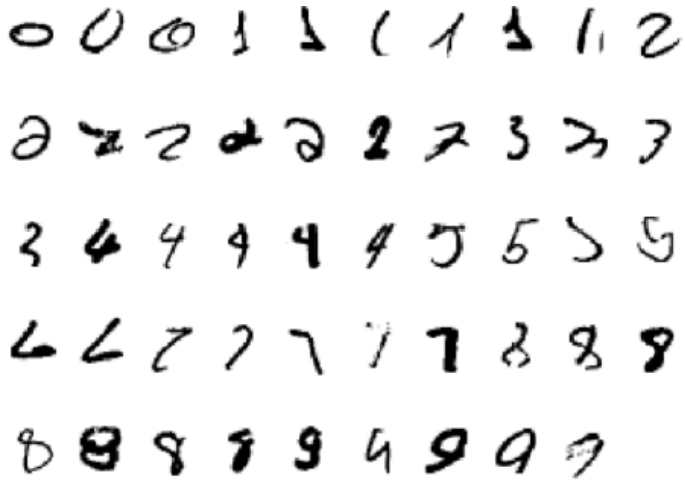
Learned classifier
$f(\boldsymbol{x})$

Training output labels $y_i \in \{0,1,\ldots,9\}$

- Do not use your "expert" knowledge
- Learn the function from data!
- Supervised learning:
  - Get many labeled examples $(\boldsymbol{x}_i, y_i), i = 1,\ldots,N$ (Called the training data)
  - Each example has an input $\boldsymbol{x}_i$ and output y_i
  - Learn a function $f(\boldsymbol{x})$ such that: $f(\boldsymbol{x}_i) = y_i$ for "most" training examples

ML Approach Benefits and Challenges

- Learned systems do very well on image recognition problems
  - On MNIST, current systems get <0.21% errors (as of 1/20/2018)
  - Used widely in commercial systems today (e.g. OCR)
  - Cannot match this performance with an expert system



- But there are challenges:
  - How do we acquire data? Someone has to manually label examples.
  - How do we parametrize a set of functions $f(\boldsymbol{x})$ to search?
  - How do we fit the function to data?
  - If a function works on training example, will it generalize on new data?

- This is what you will learn in this class