Image Classification

Special thanks to
Janine Tiefenbruck
and
Marina Langlois

Motivation

- Task: assign an input image one label from a fixed set of categories

- Why do you think it might be an important problem to solve?

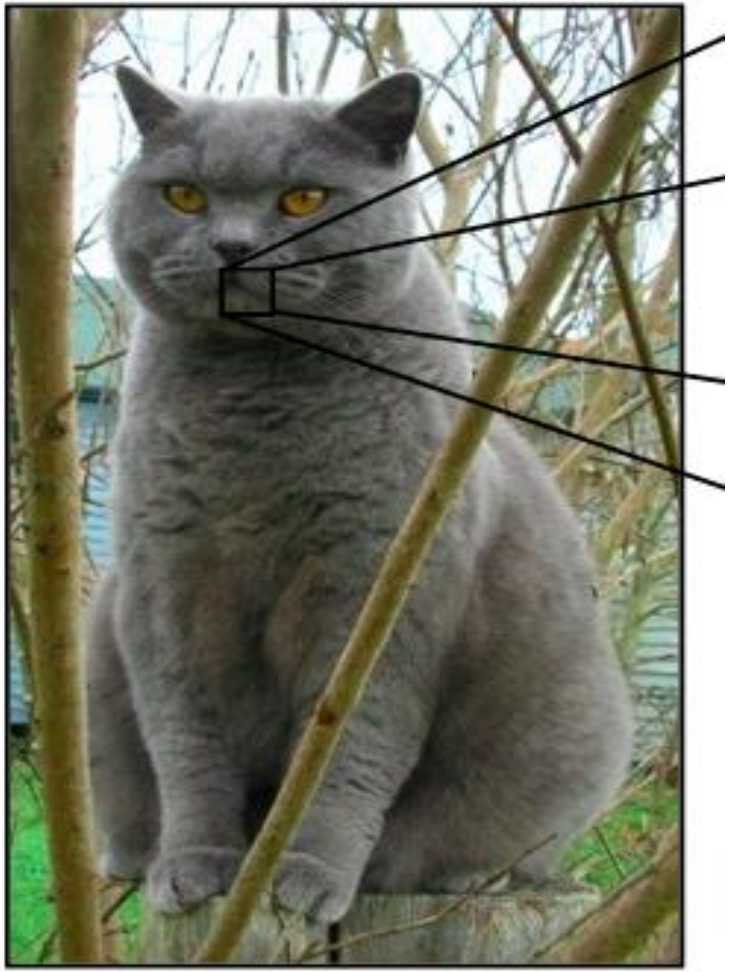
Image classification example

Ignore the lines for now.

- image classification model takes an image and assigns probabilities to 4 labels *{cat, dog, hat, mug}*

- image is represented as one large 2-dimensional array of numbers + information about the colors

- image is 248 pixels wide, 400 pixels tall, and has three color channels **Red**, **Green**, **Blue**



http://cs231n.github.io/classification/

Image classification example

Image is 248 pixels wide, 400 pixels tall, and has three color channels **Red**, **Green**, **Blue**

The image consists of:

A: 248 + 400 + 3 numbers

B: 248 * 400 * 3 numbers

C: (248 * 400) ** 3 numbers

D: (248 + 400) ** 3 numbers

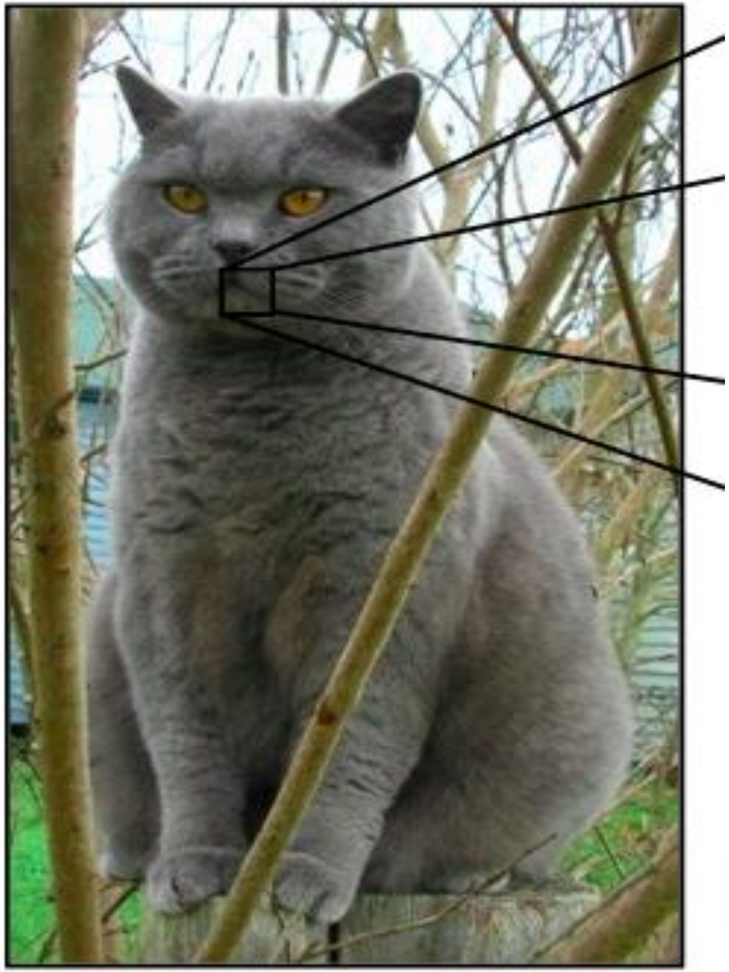
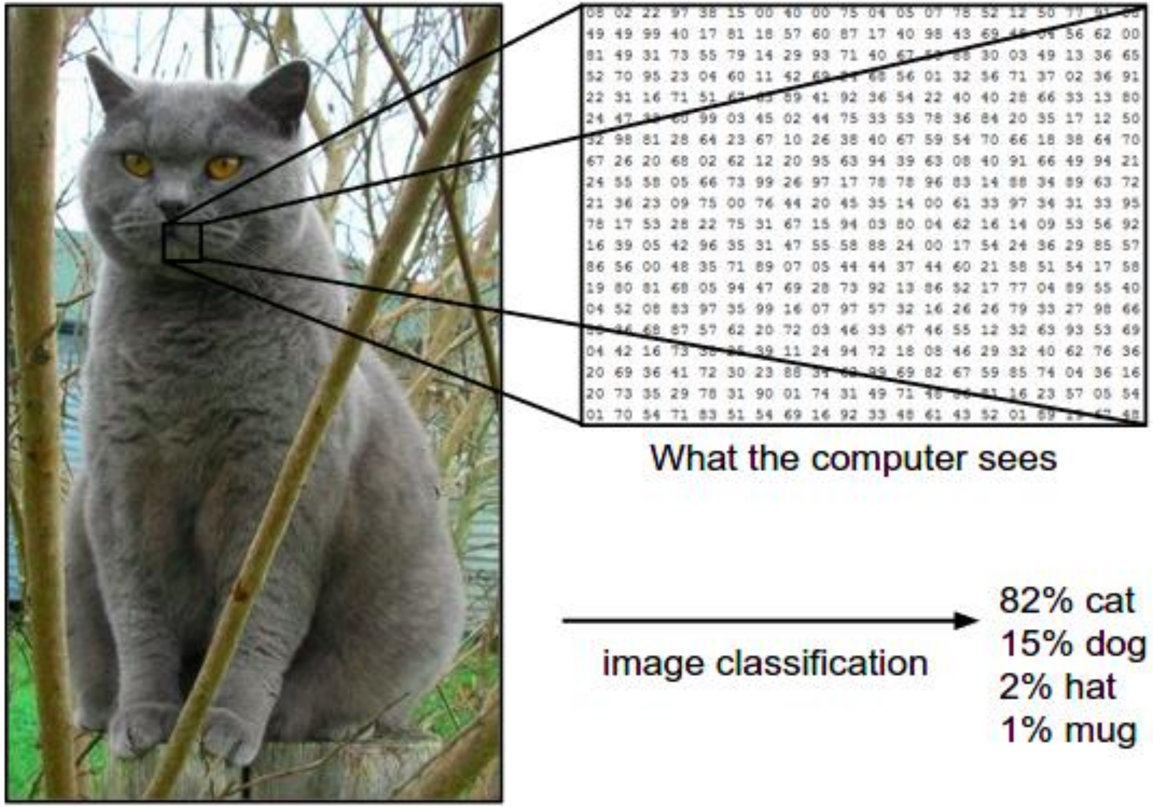


Image Classification Example



```
08 02 22 97 38 15 00 40 00 75 04 05 07 78 52 12 50 77 91 08
49 49 99 40 17 81 18 57 60 87 17 40 98 43 69 48 04 56 62 00
81 49 31 73 55 79 14 29 93 71 40 67 53 88 30 03 49 13 36 65
52 70 95 23 04 60 11 42 69 24 68 56 01 32 56 71 37 02 36 91
22 31 16 71 51 67 63 89 41 92 36 54 22 40 40 28 66 33 13 80
24 47 32 60 99 03 45 02 44 75 33 53 78 36 84 20 35 17 12 50
32 98 81 28 64 23 67 10 26 38 40 67 59 54 70 66 18 38 64 70
67 26 20 68 02 62 12 20 95 63 94 39 63 08 40 91 66 49 94 21
24 55 58 05 66 73 99 26 97 17 78 78 96 83 14 88 34 89 63 72
21 36 23 09 75 00 76 44 20 45 35 14 00 61 33 97 34 31 33 95
78 17 53 28 22 75 31 67 15 94 03 80 04 62 16 14 09 53 56 92
16 39 05 42 96 35 31 47 55 58 88 24 00 17 54 24 36 29 85 57
86 56 00 48 35 71 89 07 05 44 44 37 44 60 21 58 51 54 17 58
19 80 81 68 05 94 47 69 28 73 92 13 86 52 17 77 04 89 55 40
04 52 08 83 97 35 99 16 07 97 57 32 16 26 26 79 33 27 98 66
88 36 68 87 57 62 20 72 03 46 33 67 46 55 12 32 63 93 53 69
04 42 16 73 38 25 39 11 24 94 72 18 08 46 29 32 40 62 76 36
20 69 36 41 72 30 23 88 34 62 99 69 82 67 59 85 74 04 36 16
20 73 35 29 78 31 90 01 74 31 49 71 48 86 81 16 23 57 05 54
01 70 54 71 83 51 54 69 16 92 33 48 61 43 52 01 89 19 67 48
```

What the computer sees

image classification

82% cat
15% dog
2% hat
1% mug

Challenges

Can you think of any challenges in classifying images?

(from a computer’s perspective)

Challenges

- **Viewpoint variation.**
- **Scale variation.**
- **Deformation**.
- **Occlusion.** The object of interest can be hidden. Sometimes only a small portion of an object (as little as few pixels) could be visible.
- **Illumination conditions.**
- **Background clutter**. The objects of interest may blend into their environment, making them hard to identify.
- **Intra-class variation.** The classes of interest can often be relatively broad.

http://cs231n.github.io/classification/

Guess

- **Viewpoint variation.**
- **Scale variation.**
- **Deformation**.
- **Occlusion.**
- **Illumination conditions.**
- **Background clutter**.
- **Intra-class variation.**

Guess

- **Viewpoint variation.**
- **Scale variation.**
- **Deformation**.
- **Occlusion.**
- **Illumination conditions.**
- **Background clutter**.
- **Intra-class variation.**

Guess

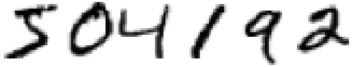
- **Viewpoint variation.**
- **Scale variation.**
- **Deformation**.
- **Occlusion.**
- **Illumination conditions.**
- **Background clutter**.
- **Intra-class variation.**

Guess

- **Viewpoint variation.**
- **Scale variation.**
- **Deformation**.
- **Occlusion.**
- **Illumination conditions.**
- **Background clutter**.
- **Intra-class variation.**

Another related problem: recognizing handwritten digits

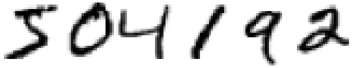
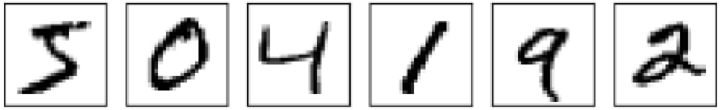
- Easy, right? How did you do it? Could you describe a “9” to me?



- But not for a computer.

- Application: sorting mail

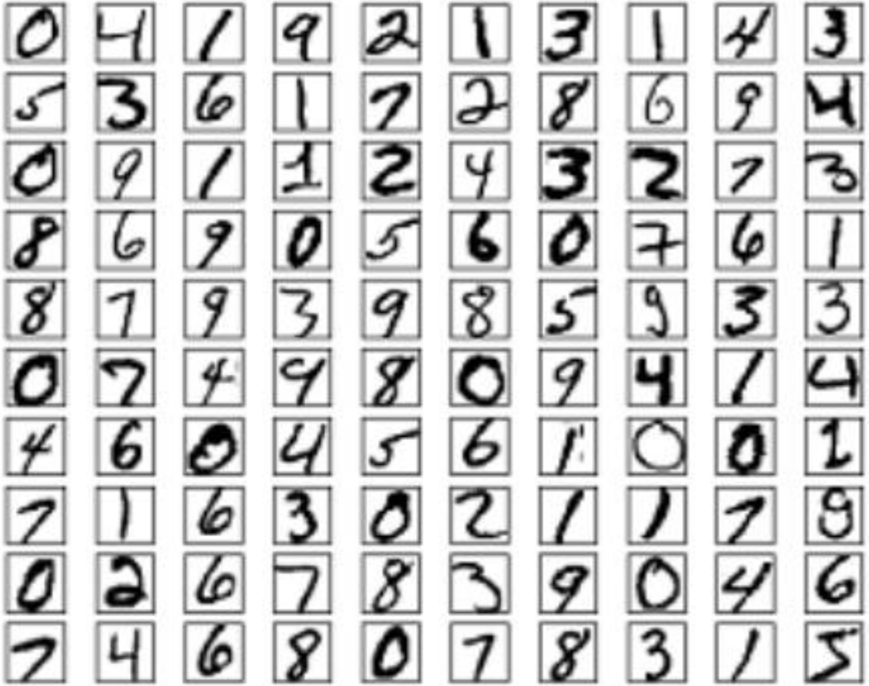
Segmentation

- Convert  
- Then recognize an individual digit.

- It is worth solving the second problem first: recognizing individual digits.
- Once solved, segmentation problem can be solved as well:
    - One approach: try many ways to segment an image and score each trial
    - A trial segmentation gets a high score if the individual digit classifier is confident of its classification in all segments

Approach to classify individual digits

- take a large number of handwritten digits with labels. (Someone labeled them already).
- This is a training set:



Approach to solve it

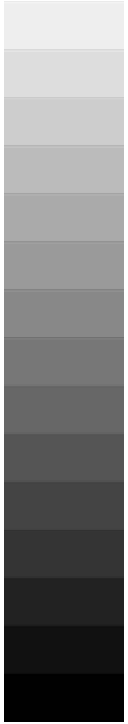
- Develop a system which can *learn* from those training examples.
- By increasing the number of training examples, the network can learn more about handwriting, and so improve its accuracy.
    - Thousands, millions..

We have an image with a number. What is next?

- We need to convert an image to digits. How?

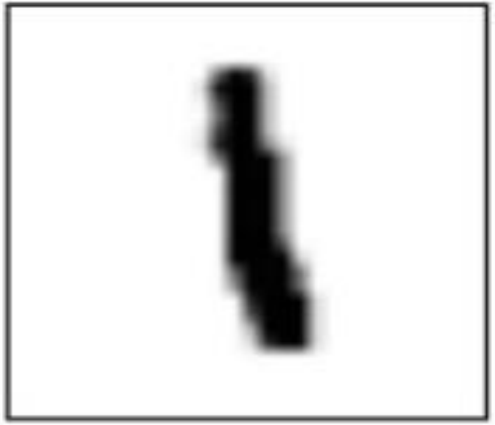
- Let’s say we have just black and white image.

- Now how about greyscale?



We have an image with a number. What’s next?

MNIST Example (Predict the Number in the Image)



$$\approx \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & .6 & .8 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & .7 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & .7 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & .5 & 1 & .4 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & .4 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & .4 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & .7 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & .9 & 1 & .1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & .3 & 1 & .1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}$$

source: https://www.tensorflow.org/versions/master/tutorials/mnist/beginners/index.html

Training and Testing Sets

- MNIST (Modified National Institute of Standards and Technology) database
- Contains 60,000 training images and 10,000 testing images

Not every image can be placed in the database.

- Each image is 28 pixels by 28 pixels (total 784 numbers)
- Associated with a digit between 0-9.
- Anti - aliased (minimizing blocky patterns, when representing a high-resolution signal at a lower resolution, link to see an example)

https://en.wikipedia.org/wiki/MNIST_database

Artificial Neural Networks

- As a model of human brain

Neural networks can

- identify faces,
- recognize speech,
- read your handwriting,
- translate texts,
- play games (typically board games or card games)
- control autonomous vehicles and robots

https://appliedgo.net/perceptron/

Today’s Exercise

Implement an algorithm that gives some evidence about whether an image has been photoshopped.

Details in Jupyter notebook in GitHub.