# CSC321 Lecture 12: Image Classification

Roger Grosse

# Midterm

- Tuesday, Feb. 28, during class
- 50 minutes
- What you're responsible for:
    - Lectures, up through L12 (this one)
    - Tutorials, up through T6 (this week)
    - Weekly homeworks, up through HW6
    - Programming assignments, up through PA2
- Emphasis on concepts covered in multiple of the above
- There will be some conceptual questions and some mathematical questions (similar to individual steps of the weekly homeworks)
- No formal proofs necessary, but you should justify your answers.
- Practice exams will be posted.

# Mid-Course Survey

Please take 10 minutes to fill out the mid-course survey.

# Overview

- Object recognition is the task of identifying which object category is present in an image.
- It's challenging because objects can differ widely in position, size, shape, appearance, etc., and we have to deal with occlusions, lighting changes, etc.
- Why we care about it
  - Direct applications to image search
  - Closely related to object detection, the task of locating all instances of an object in an image
    - E.g., a self-driving car detecting pedestrians or stop signs
- For the past 5 years, all of the best object recognizers have been various kinds of conv nets.

# Recognition Datasets

- In order to train and evaluate a machine learning system, we need to collect a dataset. The design of the dataset can have major implications.
- Some questions to consider:
  - Which categories to include?
  - Where should the images come from?
  - How many images to collect?
  - How to normalize (preprocess) the images?

# Image Classification

- Conv nets are just one of many possible approaches to image classification. However, they have been by far the most successful for the last 5 years.
- Biggest image classification "advances" of the last two decades
  - Datasets have gotten much larger (because of digital cameras and the Internet)
  - Computers got much faster
    - Graphics processing units (GPUs) turned out to be really good at training big neural nets; they're generally about 30 times faster than CPUs.
  - As a result, we could fit bigger and bigger neural nets.

# MNIST Dataset

- MNIST dataset of handwritten digits
  - **Categories:** 10 digit classes
  - **Source:** Scans of handwritten zip codes from envelopes
  - **Size:** 60,000 training images and 10,000 test images, grayscale, of size $28 \times 28$
  - **Normalization:** centered within in the image, scaled to a consistent size
    - The assumption is that the digit recognizer would be part of a larger pipeline that segments and normalizes images.
- In 1998, Yann LeCun and colleagues built a conv net called LeNet which was able to classify digits with 98.9% test accuracy.
  - It was good enough to be used in a system for automatically reading numbers on checks.

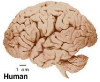
# Caltech101

- Caltech101 was the first major object recognition dataset, collected in 2003.
- Design decisions:
  - **Categories:** 101 object categories; open the dictionary to random pages and select from nouns which were associated with images
  - **Source:** find candidates with Google Image Search, hand-select the ones that actually represent the object category
  - **Number of examples:** as many as possible per category
    - most machine learning benchmarking is done using a fixed number of training examples per category (usually between 1 and 20)
  - **Normalization:**
    - Scale to be 300 pixels wide
    - Flip so that object is facing the same direction
    - Rotate certain object categories because their proposed algorithm couldn't handle vertical objects

# Caltech101

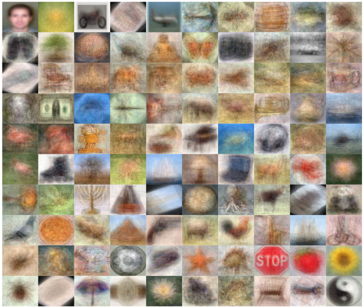




# Caltech101

- Beware of dataset biases. These are idiosyncrasies of a dataset resulting from how it was collected or normalized.
- An algorithm can appear to have good training and test error, but fail to generalize if the training data doesn't resemble the real world.
- E.g., here are the averages of all the images from some of the categories. The sizes and locations are a lot more regular than you would expect in the "real world."



# Caltech101

- There was lots of work on Caltech101 for 5 years or so, but it quickly became clear that dataset biases made it too gameable.
- By contrast, MNIST is still a productive source of insights 20 years after its introduction!

# ImageNet

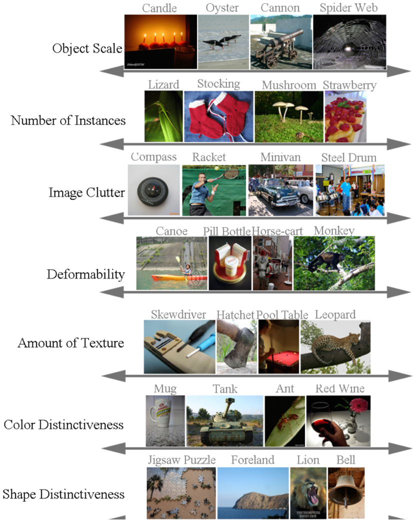
ImageNet is the modern object recognition benchmark dataset. It was introduced in 2009, and has led to amazing progress in object recognition since then.

# ImageNet

- Used for the ImageNet Large Scale Visual Recognition Challenge (ILSVRC), an annual benchmark competition for object recognition algorithms
- Design decisions
  - **Categories:** Taken from a lexical database called WordNet
    - WordNet consists of "synsets", or sets of synonymous words
    - They tried to use as many of these as possible; almost 22,000 as of 2010
    - Of these, they chose the 1000 most common for the ILSVRC
    - The categories are really specific, e.g. hundreds of kinds of dogs
  - **Size:** 1.2 million full-sized images for the ILSVRC
  - **Source:** Results from image search engines, hand-labeled by Mechanical Turkers
    - Labeling such specific categories was challenging; annotators had to be given the WordNet hierarchy, Wikipedia, etc.
  - **Normalization:** none, although the contestants are free to do preprocessing

# ImageNet

Images and object categories vary on a lot of dimensions




Russakovsky et al.

# ImageNet

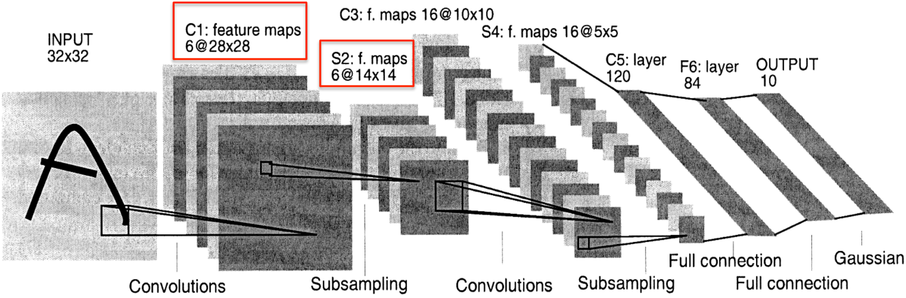
Size on disk:

MNIST
60 MB


ImageNet
50 GB

# LeNet

Here's the LeNet architecture, which was applied to handwritten digit recognition on MNIST in 1998:



# Size of a Conv Net

- Ways to measure the size of a network:
  - **Number of units.** This is important because

# Size of a Conv Net

- Ways to measure the size of a network:
  - **Number of units.** This is important because the activations need to be stored in memory during training (i.e. backprop).

# Size of a Conv Net

- Ways to measure the size of a network:
  - **Number of units.** This is important because the activations need to be stored in memory during training (i.e. backprop).
  - **Number of weights.** This is important because

# Size of a Conv Net

- Ways to measure the size of a network:
  - **Number of units.** This is important because the activations need to be stored in memory during training (i.e. backprop).
  - **Number of weights.** This is important because the weights need to be stored in memory, and because the number of parameters determines the amount of overfitting.

# Size of a Conv Net

- Ways to measure the size of a network:
  - **Number of units.** This is important because the activations need to be stored in memory during training (i.e. backprop).
  - **Number of weights.** This is important because the weights need to be stored in memory, and because the number of parameters determines the amount of overfitting.
  - **Number of connections.** This is important because

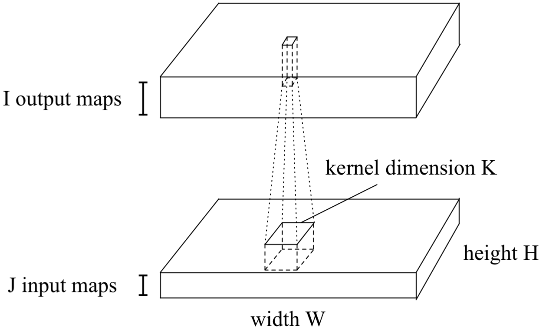
# Size of a Conv Net

- Ways to measure the size of a network:
  - **Number of units.** This is important because the activations need to be stored in memory during training (i.e. backprop).
  - **Number of weights.** This is important because the weights need to be stored in memory, and because the number of parameters determines the amount of overfitting.
  - **Number of connections.** This is important because there are approximately 3 add-multiply operations per connection (1 for the forward pass, 2 for the backward pass).

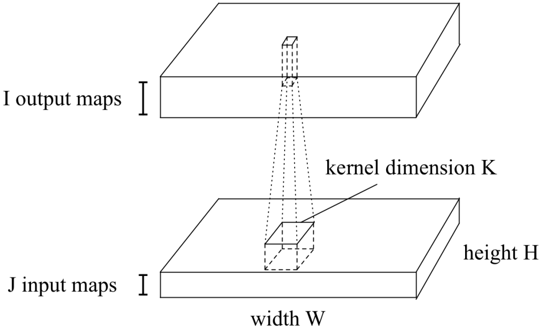
# Size of a Conv Net

- Ways to measure the size of a network:
  - **Number of units.** This is important because the activations need to be stored in memory during training (i.e. backprop).
  - **Number of weights.** This is important because the weights need to be stored in memory, and because the number of parameters determines the amount of overfitting.
  - **Number of connections.** This is important because there are approximately 3 add-multiply operations per connection (1 for the forward pass, 2 for the backward pass).
- We saw that a fully connected layer with $M$ input units and $N$ output units has $MN$ connections and $MN$ weights.
- The story for conv nets is more complicated.

# Size of a Conv Net

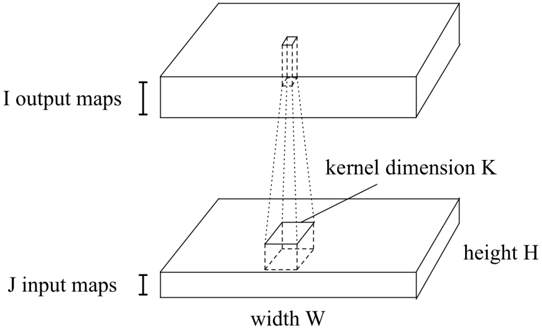


# Size of a Conv Net



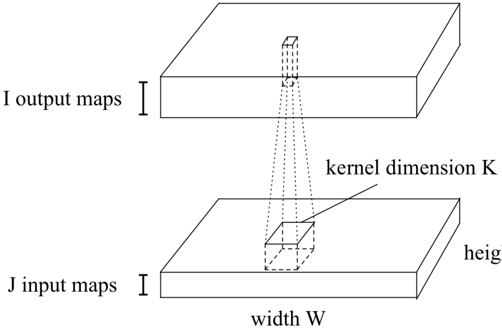

| | **fully connected layer** | **convolution layer** |
|---|---|---|
| **# output units** | | |

# Size of a Conv Net



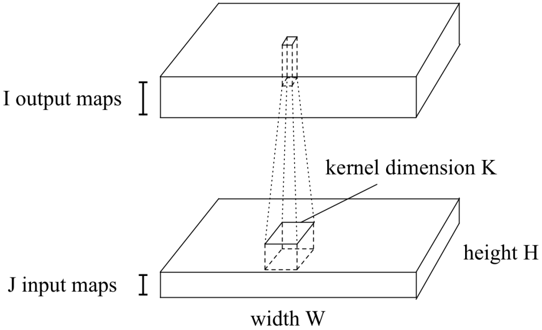

| | fully connected layer | convolution layer |
|---|---|---|
| **# output units** | $WHI$ | $WHI$ |

# Size of a Conv Net



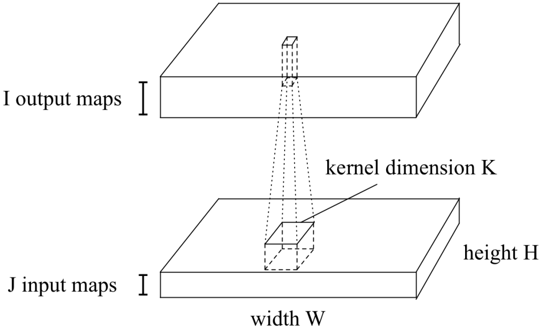

| | **fully connected layer** | **convolution layer** |
|---|---|---|
| **# output units** | *WHI* | *WHI* |
| **# weights** | | |

# Size of a Conv Net



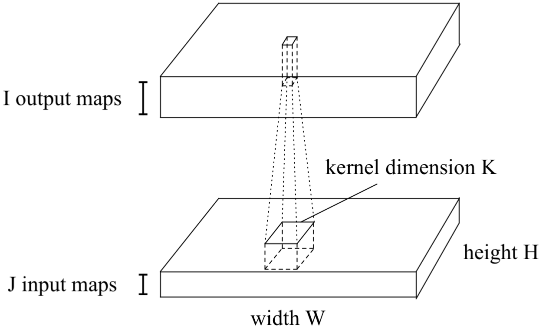

| | fully connected layer | convolution layer |
|---|---|---|
| **# output units** | $WHI$ | $WHI$ |
| **# weights** | $W^2H^2IJ$ | |

# Size of a Conv Net



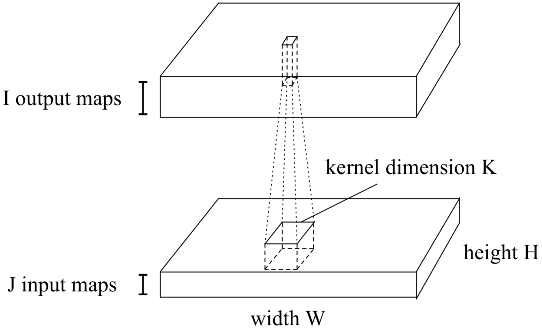

| | **fully connected layer** | **convolution layer** |
|---|---|---|
| **# output units** | $WHI$ | $WHI$ |
| **# weights** | $W^2H^2IJ$ | $K^2IJ$ |

# Size of a Conv Net



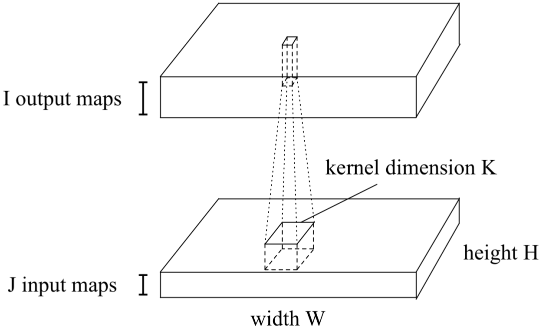

| | fully connected layer | convolution layer |
|---|---|---|
| # output units | $WHI$ | $WHI$ |
| # weights | $W^2H^2IJ$ | $K^2IJ$ |
| # connections | | |

# Size of a Conv Net




| | fully connected layer | convolution layer |
|---|---|---|
| **# output units** | $WHI$ | $WHI$ |
| **# weights** | $W^2H^2IJ$ | $K^2IJ$ |
| **# connections** | $W^2H^2IJ$ | |

# Size of a Conv Net




| | fully connected layer | convolution layer |
|---|---|---|
| # output units | $WHI$ | $WHI$ |
| # weights | $W^2H^2IJ$ | $K^2IJ$ |
| # connections | $W^2H^2IJ$ | $WHK^2IJ$ |

# Size of a Conv Net

Sizes of layers in LeNet:

| Layer | Type | # units | # connections | # weights |
|---|---|---|---|---|
| C1 | convolution | 4704 | 117,600 | 150 |
| S2 | pooling | 1176 | 4704 | 0 |
| C3 | convolution | 1600 | 240,000 | 2400 |
| S4 | pooling | 400 | 1600 | 0 |
| F5 | fully connected | 120 | 48,000 | 48,000 |
| F6 | fully connected | 84 | 10,080 | 10,080 |
| output | fully connected | 10 | 840 | 840 |

Conclusions?

# Size of a Conv Net

- Rules of thumb:
  - Most of the units and connections are in the convolution layers.
  - Most of the weights are in the fully connected layers.
- If you try to make layers larger, you'll run up against various resource limitations (i.e. computation time, memory)
- Conv nets have gotten a LOT larger since 1998!

# Size of a Conv Net

| | **LeNet (1989)** | **LeNet (1998)** | **AlexNet (2012)** |
|---|---|---|---|
| **classification task** | digits | digits | objects |

# Size of a Conv Net

| | **LeNet (1989)** | **LeNet (1998)** | **AlexNet (2012)** |
|---|---|---|---|
| **classification task** | digits | digits | objects |
| **categories** | 10 | 10 | 1,000 |

# Size of a Conv Net

| | **LeNet (1989)** | **LeNet (1998)** | **AlexNet (2012)** |
|---|---|---|---|
| **classification task** | digits | digits | objects |
| **categories** | 10 | 10 | 1,000 |
| **image size** | $16 \times 16$ | $28 \times 28$ | $256 \times 256 \times 3$ |

# Size of a Conv Net

| | LeNet (1989) | LeNet (1998) | AlexNet (2012) |
|---|---|---|---|
| **classification task** | digits | digits | objects |
| **categories** | 10 | 10 | 1,000 |
| **image size** | $16 \times 16$ | $28 \times 28$ | $256 \times 256 \times 3$ |
| **training examples** | 7,291 | 60,000 | 1.2 million |

# Size of a Conv Net

| | **LeNet (1989)** | **LeNet (1998)** | **AlexNet (2012)** |
|---|---|---|---|
| **classification task** | digits | digits | objects |
| **categories** | 10 | 10 | 1,000 |
| **image size** | $16 \times 16$ | $28 \times 28$ | $256 \times 256 \times 3$ |
| **training examples** | 7,291 | 60,000 | 1.2 million |
| **units** | 1,256 | 8,084 | 658,000 |

# Size of a Conv Net

| | **LeNet (1989)** | **LeNet (1998)** | **AlexNet (2012)** |
|---|---|---|---|
| **classification task** | digits | digits | objects |
| **categories** | 10 | 10 | 1,000 |
| **image size** | $16 \times 16$ | $28 \times 28$ | $256 \times 256 \times 3$ |
| **training examples** | 7,291 | 60,000 | 1.2 million |
| **units** | 1,256 | 8,084 | 658,000 |
| **parameters** | 9,760 | 60,000 | 60 million |

# Size of a Conv Net

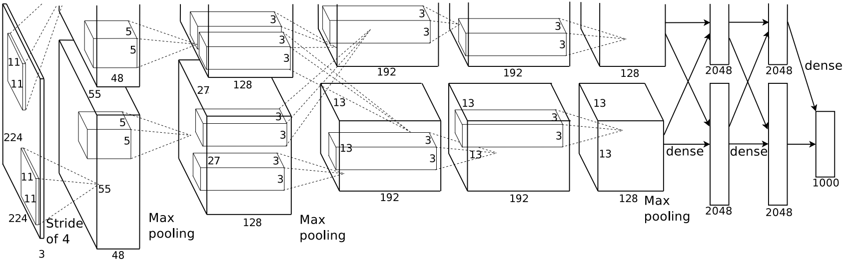
| | **LeNet (1989)** | **LeNet (1998)** | **AlexNet (2012)** |
|---|---|---|---|
| **classification task** | digits | digits | objects |
| **categories** | 10 | 10 | 1,000 |
| **image size** | $16 \times 16$ | $28 \times 28$ | $256 \times 256 \times 3$ |
| **training examples** | 7,291 | 60,000 | 1.2 million |
| **units** | 1,256 | 8,084 | 658,000 |
| **parameters** | 9,760 | 60,000 | 60 million |
| **connections** | 65,000 | 344,000 | 652 million |

# Size of a Conv Net

| | LeNet (1989) | LeNet (1998) | AlexNet (2012) |
|---|---|---|---|
| **classification task** | digits | digits | objects |
| **categories** | 10 | 10 | 1,000 |
| **image size** | $16 \times 16$ | $28 \times 28$ | $256 \times 256 \times 3$ |
| **training examples** | 7,291 | 60,000 | 1.2 million |
| **units** | 1,256 | 8,084 | 658,000 |
| **parameters** | 9,760 | 60,000 | 60 million |
| **connections** | 65,000 | 344,000 | 652 million |
| **total operations** | 11 billion | 412 billion | 200 quadrillion (est.) |

# AlexNet

- AlexNet, 2012. 8 weight layers. 16.4% top-5 error (i.e. the network gets 5 tries to guess the right category).




(Krizhevsky et al., 2012)

- They used lots of tricks we've covered in this course (ReLU units, weight decay, data augmentation, SGD with momentum, dropout)
- AlexNet's stunning performance on the ILSVRC is what set off the deep learning boom of the last 5 years.

# GoogLeNet

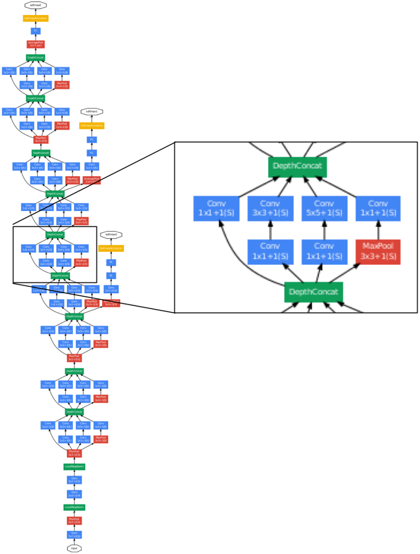
GoogLeNet, 2014.

22 weight layers

Fully convolutional (no fully connected layers)

Convolutions are broken down into a bunch of smaller convolutions

6.6% test error on ImageNet



# GoogLeNet

- They were really aggressive about cutting the number of parameters.
  - Motivation: train the network on a large cluster, run it on a cell phone
    - Memory at test time is the big constraint.
    - Having lots of units is OK, since the activations only need to be stored at training time (for backpropagation).
    - Parameters need to be stored both at training and test time, so these are the memory bottleneck.
  - How they did it
    - No fully connected layers (remember, these have most of the weights)
    - Break down convolutions into multiple smaller convolutions (since this requires fewer parameters total)
  - GoogLeNet has "only" 2 million parameters, compared with 60 million for AlexNet
  - This turned out to improve generalization as well. (Overfitting can still be a problem, even with over a million images!)

# Classification

ImageNet results over the years. Note that errors are top-5 errors (the network gets to make 5 guesses).

| Year | Model | Top-5 error |
|---|---|---|
| 2010 | Hand-designed descriptors + SVM | 28.2% |
| 2011 | Compressed Fisher Vectors + SVM | 25.8% |
| 2012 | AlexNet | 16.4% |
| 2013 | a variant of AlexNet | 11.7% |
| 2014 | GoogLeNet | 6.6% |
| 2015 | deep residual nets | 4.5% |

We'll cover deep residual nets later in the course, since they require an idea we haven't covered yet.

Human-performance is around 5.1%.

They stopped running the object recognition competition because the performance is already so good.