

Image Classification

Reading

- Deep Learning: chapter 12.2
 - <https://www.deeplearningbook.org/contents/applications.html>
- Very brief overview, will go into a bit more depth in the slides

Large-Scale Training

- There is a widely held belief in deep learning that bigger models are better
 - Occam's Razor (informal): you should always use the simplest model that can solve your task
 - Increasing the model complexity is rarely the solution to training issues
- Before adding more neurons, try to understand why training is not working
 - Is the data noisy, properly normalized, correctly labeled?
 - Are other hyperparameters (learning rate, loss function) reasonably chosen?
 - Are there bugs in the code (very often the case for beginners!)?

Common carbon footprint benchmarks

in lbs of CO2 equivalent

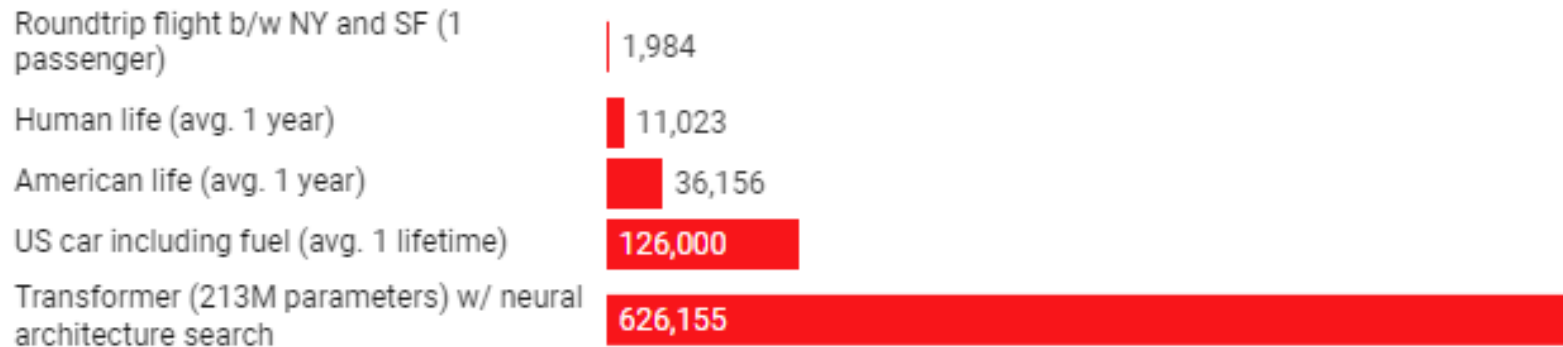


Chart: MIT Technology Review • Source: Strubell et al. • Created with Datawrapper

ChatGPT's electricity consumption (not including training costs) is equivalent to that of up to 175,009 Danes!

(source: <https://medium.com/@chriscointon/the-carbon-footprint-of-chatgpt-e1bc14e4cc2a>)

Computer Vision

- One of the classical ML tasks
- Very easy for humans
 - One of the building blocks of intelligence
- Relatively easy to collect tons of labeled data
 - Many, many datasets and competitions
- Many applications
 - Robotics
 - Medical imaging
 - Mapping
 - Face recognition

Image Data

- Grayscale or color images
- A grayscale image is just a matrix of integers between $[0, 255]$
 - where 0 is black, 255 is white, in between is (shades of) grey
- A color image is also a matrix, but each entry has three values (“channels”)
 - RGB – red, green, blue
 - where $[0, 0, 0]$ is black; $[255, 255, 255]$ is white; $[255, 0, 0]$ is red, etc.
- Note: each matrix is “flattened” before being sent to a fully-connected layer since those only accept vectors

Data Preprocessing

- Normalization – normalize data to some reasonable range
 - Typically $[-1, 1]$, $[-0.5, 0.5]$, $[0, 1]$
 - Very common across all ML tasks
 - Very important for images since pixels range from 0 to 255
 - Large ranges destabilize gradient descent
 - Usually just subtract the data mean and divide by range, though normalization could be learned also
- Contrast normalization
 - Subtract the mean or reduce the contrast in some other way
 - Less common than standard normalization

Data Augmentation

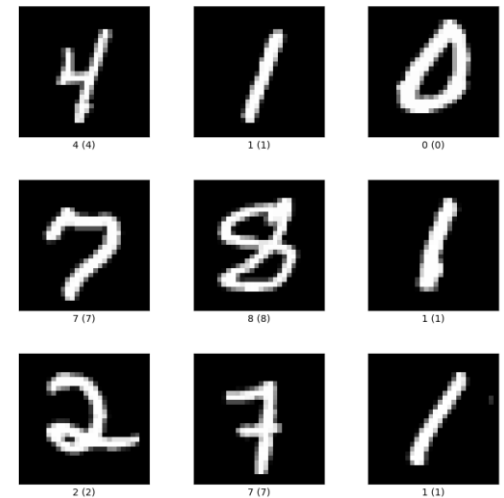
- One of the most important tricks in image classification
- Increase dataset size by adding small perturbations to images
 - Small rotations, translations, stretching, noise
- Statistically, this is not justified
 - Data no longer IID, it is essentially made up
 - It is very effective, however
 - Notion of distribution in images is quite blurry
 - Keep in mind deep learning is as much about optimization as it is about statistical learning

Standard Datasets

- Many datasets out there
- General datasets with many classes (cats, dogs, cars)
 - CIFAR, ImageNet
- Specialized datasets
 - Handwritten digits – MNIST
 - Office
 - Street signs
 - Birds
 - Video datasets

MNIST

- The “Hello, world” dataset of ML
- Relatively easy dataset that is still quite useful as a toy-but-not-so-toy dataset
- Grayscale 28×28 images
 - 10 classes of handwritten images
 - 60,000 training images (6K per class)
 - 10,000 test images (1K per class)
- An easy dataset because most pixels are either black or white
 - You can solve it with SVMs also
 - Even small NNs achieve 95+% accuracy



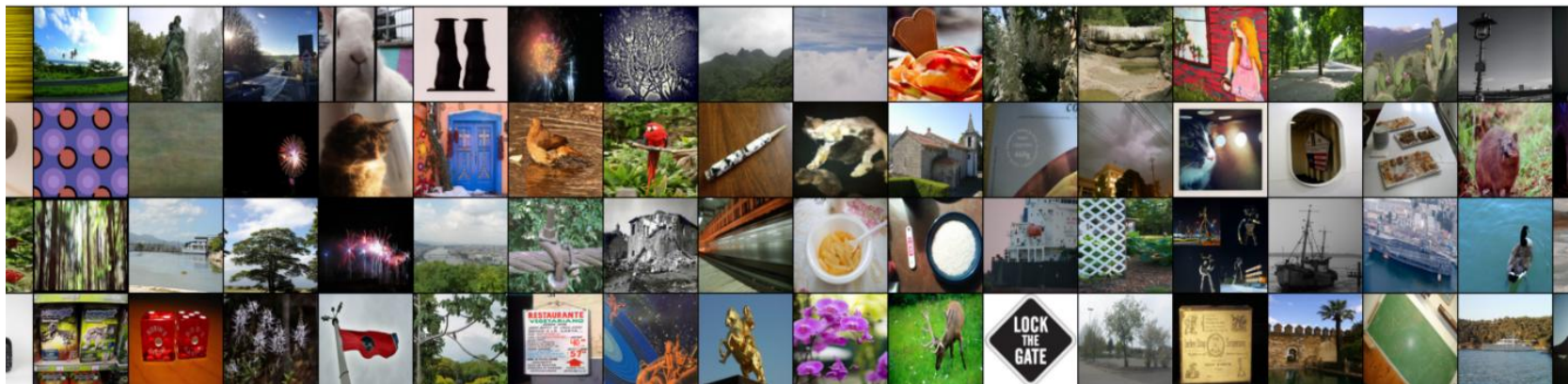
CIFAR

- Deceptively hard dataset
- $32 \times 32 \times 3$ color images of everyday objects
 - Birds, cars, planes, etc.
 - 10-class and 100-class versions
- Basic architectures achieve ~80% accuracy on CIFAR-10
 - Need larger architectures to get to the 90s
 - CIFAR-100 is even harder
- Challenge is that images are too rich but have too few pixels



ImageNet

- One of the most popular and challenging datasets
 - More than 14m images and 21 thousand classes
- $224 \times 224 \times 3$ color images of everyday objects
 - Classes very specific, sometimes it's hard to distinguish
- Very hard dataset
 - Sometimes multiple objects, occlusion, etc.
 - Need very large models/a lot of computation to get to ~90% accuracy
 - Even humans achieve only ~95% accuracy



RPI Buildings

- Charles Yu '23 and I collected images of buildings on campus
 - One building per image
 - Different time of day, weather conditions, distance
- 11 buildings: Lally, Sage, Troy, 87gym, library, Ricketts, Voorhees, Greene, JEC, Amos Eaton, Empac
 - about 500 images per building
- Each image is $4000 \times 3000 \times 3$
 - Probably needs to be shrunk for easier learning

