Literature Review: ImageNet Classification with Deep Convolutional Neural Networks

Yufeng Yuan

CS281K Computational Photography

Nov.30th 2017

Outline

Abstract

- Dataset: 1.2 million images, 1000 different classes
- Model: a deep convolutional neural network
- Tricks for training: RELU, GPU implementation
- Tricks for overfitting: dropout
- Result: considerably better than the previous state-of-the-art

Examples

Some examples of commonly used commands and features are included, to help you get started.

Introduction

Machine learning methods already perform well on simple object recognition with small dataset, but it has only recently become possible to collect labeled datasets with millions of images for recognition in realistic settings.
Models with large learning capacity are needed to realistic object recognition. Convolutional neural networks, whose capacity can be controlled by varying depth and breadth is one of such models with correct prior knowledge.
Convolutional neural networks have still been prohibitively expensive to aply in large scale to high-resolution images Current GPUs, paired with a highly-optimized implementation of 2D convolution, are powerful enough for the training.

Tables and Figures

- Use `tabular` for basic tables — see Table 1, for example.
- You can upload a figure (JPEG, PNG or PDF) using the files menu.
- To include it in your document, use the `includegraphics` command (see the comment below in the source code).

Item	Quantity
Widgets	42
Gadgets	13

Table 1: An example table.

Readable Mathematics

Let $X_1, X_2, \ldots, X_n$ be a sequence of independent and identically distributed random variables with $\mathsf{E}[X_i] = \mu$ and $\mathsf{Var}[X_i] = \sigma^2 < \infty$, and let

$$S_n = \frac{X_1 + X_2 + \cdots + X_n}{n} = \frac{1}{n} \sum_{i}^{n} X_i$$

denote their mean. Then as n approaches infinity, the random variables $\sqrt{n}(S_n - \mu)$ converge in distribution to a normal $\mathcal{N}(0, \sigma^2)$.