Sparsity and Structured Matrices

CS6787 Lecture 14 — Fall 2017
Sparsity Basics

• A sparse matrix has most of its entries zero
  • The fraction of nonzero entries is called the **density**

• One way to make linear algebras operations faster
  • Why does this help?

• But, it’s not that simple
  • There are many **pros** to sparse computing for ML systems
  • But there are also a lot of **cons**
With Sparsity, Storage Matters!

• Unlike dense matrices, **many different ways** to store a sparse matrix
  • COO — coordinate list
  • CSR — compressed sparse row
  • CSC — compressed sparse column

• **What are the advantages and disadvantages of these?**
Demo
General rule of thumb for performance

• For **fixed vector dimension**
  • As density decreases, cost of computations **goes down**
  • But only starts being better than dense at around 10% for many operations

• For **fixed size of data** (measured in bytes)
  • As density decreases, cost of computations **goes up**
  • In the limit of extreme sparsity, you start using techniques from databases
    • Or from graph computation
Where do we find sparsity in ML?

• In **input** training sets
  - Many real-world phenomena are sparse. **Examples?**

• In **models** that we learn
  - Particularly when we use **L1 regression**
  - Also sometimes want to **impose sparsity** on our models a priori
  - Intuition: sparse models **less prone to overfitting**

• In **intermediate values** used during computation
  - For example, the output of a **ReLU** activation function is typically sparse
Two Strategies for Leveraging Sparsity in Data

• Use **sparse linear algebra/sparse computations**
  • Hopefully this will run faster
  • You probably already know about this

• Use an **embedding**
  • Map the sparse input data onto a **lower-dimensional dense feature vector**
  • For example, with random kernel features
  • For example, with the first layer of a deep neural network
  • For example, **word2vec**
Johnson–Lindenstrauss Transform

• One popular general embedding

• Result: given $0 < \varepsilon < 1$, $m$ points in $\mathbb{R}^D$, there is a matrix $A$ such that

$$(1 - \varepsilon) \|x - y\|^2 \leq \|Ax - Ay\|^2 \leq (1 + \varepsilon) \|x - y\|^2$$

where $A \in \mathbb{R}^{d \times D}$ and $d \approx 8\varepsilon^{-2} \log(m)$

• We can use this to project sparse vectors onto a smaller dense space
  • Then use fast dense arithmetic
Sparsity on Hardware

• The **CPU usually has the most to gain** from going sparse
  • Because it has large caches that support random access

• But **GPUs can also benefit** from sparse computation
  • For example, NVIDIA has a **cuSPARSE** sparse matrix library

• If sparsity pattern is predictable, we can design **specialized hardware**
  • But I have not seen this used in production systems
More Complex Questions
Sparsity: Storage Matters — Episode 2

• Attack of the Clones!
  • Should we store multiple copies of our sparse thing in different formats?

• What precision to use for the indices?

• Should we use blocking?

• Should we use heterogeneous formats with dense sub-blocks?

• These questions can affect performance by orders of magnitude!
Questions?
Project Report Expectations
Formatting

• Report should be at least four pages, not including references

• Report should use ICML 2017 style or a similar style
  • This is mostly to be fair about length

• Report should be structured appropriately
  • For example: abstract, introduction, related work, main results, experiments
  • Correctly formatted references page
Content — Overview

• You should have **implemented a machine learning system**
  • This entails writing some code
  • You should have some **code to submit** along with the report
    • Either as a supplemental file, or as a link to a repository

• You should have used **a technique we discussed in the course**
  • And it should be clear from the report which one you used

• You should have run throughput or wall-clock time **experiments**

• Your work should correspond to the proposal
Content — Conceptual

• The report should **summarize the problem** you are trying to solve
  • Explain why your approach is a good idea or interesting to study
  • Thesis statement clearly and concisely states the purpose of the report

• The report should fairly acknowledge **previous work**
  • And relate it to what you did

• The report should be **clear and well-written**
  • Avoid grammar/spelling/punctuation issues that make the text difficult to read.

• The report should **demonstrate knowledge/understanding** of the chosen technique beyond what we discussed in class
Content — Technical

• The main section of the report should **explain what you did**
  • And **why** you did it!

• Someone should be able to **reproduce your results** from the report

• The paper should be technically sound
  • Any **claims should be supported** by theoretical analysis or experimental results

• Evaluate both the **strengths and weaknesses** of the work
Content — Experimental

• The experiments should involve a **fair comparison**
  • In terms of systems performance, among two or more methods

• The report should **explain the experimental results**
  • Why did this happen? Was it what you expected? What does this tell us?

• The results should be **properly formatted**
  • At least one figure with a title and properly labeled axes
  • Present things graphically whenever possible
Content — Impact

• The report should discuss the impact of the results
  • What does this tell us about how we should design systems in the future?

• The report should gesture at possibilities for future work
Questions?
Structured Matrices

A whiteboard talk
Questions?

• Upcoming things
  • This was the last lecture of the semester
  • Project report due on Wednesday
  • Expect everything else to be graded by the end of the week
    • Only reviews should be left, and I’ll send out a note when those are done
    • If you sent me a submission by email, double check that it was graded
  • You can apply online to TA CS4780 next semester

• Thank you!