Large Datasets

Previously in Our Class

- Talked about SVM scaling
- Mentioned XGBoost scaling
- But just how far will these algorithms go and does it even matter?

Dataset Size

Usually refers to number of samples and not variables

- Sklearn toy datasets: 150 samples to 5,056 samples
- Sklearn real-world datasets: 400 to 4.9 million (average 20,000)
- UCI datasets up to 62,000,000

When Does Big Data Begin?

- No set definition
- Rough rule of thumb: If you cannot fit your data into memory, you have “big data”
  - Memory: aka RAM
  - RAM sizes are always increasing
  - Big data is a moving target

How Much Data in 16GB of Memory?

- 1 GB is 1 billion bytes (1 byte = 8 bits)
- Store our data as 32 or 64 bit segments of memory

$$\frac{16 \cdot 10^9 \cdot 8}{64} = 2 \cdot 10^9$$

- Thus, a typical machine can store 2–4 billion numbers (in a perfect world)

Nothing Is Perfect

- Still need the operating system
- Still need processing memory
- Need to load your program
- Data is not one dimensional
- Our 4 billion numbers get eaten up pretty fast
  - If I have 10 features, then 400 million rows is 4 billion numbers
- In short, when we reach “millions” of samples, we need some new ways to deal with all this “big data”

DataScience@SMU

Stochastic Gradient Descent

Stochastic Gradient Descent

- Variant of gradient descent
- How to optimize a regression using a basic loss function
- Quick gradient descent review

$$m_{n+1} = m_n - \alpha \nabla J(m)$$

- Or the new slopes, m, are just the old slopes subtracting the derivative of the loss function (J) times a small number (α)

Reminders

- J includes **all** the data
  - Predictions and targets
  - To calculate this, you need to calculate all your predictions and then update all your slopes at once
- How does stochastic gradient descent differ?
  - What does it give us?

Stochastic Gradient Descent

- Rather than take **all** the data, take a **sample** of the data
  - A sample is called a batch.
- Rather that use all the data, use the batch data to estimate the new slopes

$$m_{n+1} = m_n - \alpha \nabla Q(m)$$

- Where Q is an approximation of J

Advantages

- Since we no longer need **all** the data in memory, we can process much larger datasets as we “sweep” through the data, sampling the data in batches.
- If we sweep through the data once (enough batches to equal the full data size) we call that one epoch.
- This saves us huge amounts of memory since we can load a single sample at a time and process it.
- We **do** have to figure out how to get data quickly into memory because this will create a bottleneck reading data from a file.

DataScience@SMU

Hashing Trick

SGD and Out of Core

- We just learned that we can do regression by only using batches of data.
- This gives us a new problem.
  - How do we load only part of the data into memory?
  - How do we load fast enough to keep the processor busy?
    - No bottleneck
- Loading only part of the data is called “out-of-core” learning.
- Most algorithms for big data rely on out-of-core learning of some sort.

Hashing Trick

- Storing data in memory in arrays can be time-consuming and expensive.
- The hashing trick is the key to quickly make a vector in memory.
- Hash functions are functions that produce a fixed-length unique output.
- Using a hash function, the feature is mapped to a space in memory quickly.
- Memory is read as a vector.
- Any values of the feature are mapped into memory as well.

Hash Collisions

- If two columns hash to the same address, this is known as a hash collision.
- Scholarship suggests that this is not as big a problem as it might seem.
- Just make your memory space big enough for all your features.

Summary

- As a data scientist, one doesn’t do much with the hashing trick—it is just there.
- It allows you to get data into memory in batches quickly.
- It’s at the heart of most big data algorithms.

DataScience@SMU

SGD and Epochs

Using SGD and Epochs

- SGD works the same as a linear/logistic regression.
- Regularization is still a good practice.
- Adjusting the learning rate is also good practice.
- When running and training the model, the number of times the complete set of data is “seen” by the model is called “epochs.”
  - If I have 100 rows of data, and I feed them five at a time (one batch) into the model, after 20 batches I will have consumed every row in the dataset. This constitutes one epoch.
- It usually takes multiple epochs for a good fit.

DataScience@SMU