CSC290/571 Topics in Systems: Machines Learning Systems
ML Training Basics

Sreepathi Pai

Oct 21, 2024

URCS

Outline

Outline

Training Machine Learning (ML) Models

- There are a number of machine learning methods
  - Regression, Support Vector Machines, Deep Neural Networks, etc.
- Many tasks
  - Prediction
  - Classification
- But many common parts:
  - (Labeled) Training set
  - (Labeled) Test set
  - (also seen validation set)
  - Loss function

Training algorithm (English version)

- Pass an input x_i (usually a vector) through the model
- Observe the obtained output y_i' and the expected output y_i
- Repeat for all training inputs x_i
- Compute the loss function, which signifies the difference between all y_i' and y_i
- The goal is to minimize the loss function by changing the model's parameters
    - informed by the loss function
- Repeat until changes in loss function are small ("optimization")

(Informed by the loss function)

- The loss function depends on the inputs, outputs, and the parameters of the model
- Inputs and outputs are fixed, so changes in parameters lead to changes in loss function
- The notion of *gradient descent*
  - Compute the derivative of the loss function wrt the parameters
  - Update the parameters using this derivative (or "gradient") as a guide

How do we do this?

- For many early ML models
  - Models are simple
  - Derivatives are easy to compute by hand
  - Training sets are small
  - Optimization often leads to a global best

Outline

DNNs

- Deep Neural Networks consist of multiple layers
  - No longer simple, lots of parameters
- Derivatives?
  - Difficult to do by hand
- Training sets
  - Very large
- Optimization
  - May not lead to global best

Calculating Gradients

- Auto differentiation (AD)
  - Contrast with: symbolic differentiation, numeric differentiation
- More like a compiler algorithm
  - For every operation the program does, track a derivative
- Two modes:
  - Forward AD
  - Backward AD, e.g., Backpropagation

ML Program

- Nodes, edges
  - Nodes are operators
  - Edges are data flows
- Backward AD "adds" nodes and "edges" that compute the derivatives/gradient
  - Need extra memory
- Done behind the scenes by frameworks like PyTorch

Training Sets Very Large: Part 1

- Training is split up into epochs
  - Each epoch processes every input in the training set
- Each epoch is split into batches
  - Each batch is a partition of the the training set
  - Each batch forms an iteration
- Parameters updated once per iteration

Training Sets Very Large: Part 2

- Lots of parameters, complex gradient functions
- Expensive to compute gradient wrt to every input
- Stochastic Gradient Descent
  - Evaluate gradient for *a* randomly chosen input
  - Sometimes, a subset of randomly chosen inputs, aka a "mini-batch"

Training Sets Very Large: Part 3

- Distribute batch across machines
  - Data Parallel
- Synchronous
  - Weights updated using all-reduce
- Asynchronous
  - Weights sent to a parameter server for aggregation

Other Issues

- Data storage and retrieval
- Communication
- Fault tolerance

Outline

References

- Prof. Sa's course: Principles of Large-Scale Machine Learning (Fall 2023)
- Prof. Ullman's book chapter: Large Scale Machine Learning
- Google's Machine Learning Glossary: `https://developers.google.com/machine-learning/glossary`