HSD5230 High Performance Computing
Final Exam - Text Part

Miao Cai[*]

2019-05-09

1. The gradient, the Hessian, and optimization: Discuss the gradient and the Hessian as they apply to optimization in machine learning. Be sure to address (at least) the following points in your discussion. Your answer should be at least a few paragraphs long, if not longer. If you use a source to support your answer, cite it.

(a) Define the gradient and the Hessian in your own words, as they apply to optimization in machine learning. How does this relate to first and second order derivatives?

(b) How does optimization in machine learning differ from classic function optimization in mathematics?

(c) What is a local minimum versus a global minimum? What is a saddle point? Why are these problematic in ML optimization?

(d) Summarize the general steps an optimization algorithm takes (Usually the computer does this for you, so you don't code these steps)

(e) What are the tradeoffs in using gradient based optimization methods versus Hessian based methods?

(f) Why has the field typically used Newtonian (2nd order derivative) optimization on unregularized GLM models? Under what specific circumstance do these second order methods become challenging to implement in practice? Why?

(g) Why do we tend to use first order methods like SGD or RMSProp in deep learning?

- The gradient is how fast the cost function is changing, and the Hessian is how fast is the change in the cost function is changing. Knowing the gradient or the Hessian can help decide the learning rate and optimize the cost function. The gradient is equivalent to the first order derivative, while the Hessian is the second order derivative.
- In machine learning, the optimization is almost always related to high dimensional data with a lot of features, so the optimization is usually very difficult due to complex cost function and approximating the global minimum among multiple local minima. In contrast, optimization in mathematics is more often related to much fewer features and simpler cost function. Traditional mathematics is trying to find the exact analytical solution.
- A local minimum is where the gradient equals zero, but only the smallest value in a small range. A saddle point is where the gradient equals zero, but the gradients at the left and right side have the same sign (both positive or negative). The saddle point is problematic since it is neither a local minumum or global minimum, so the final optimization solution must not be a saddle point.
- The general steps an optimization algorithm takes are:
  1. Start from an initial value,
  2. Using the full or a small portion of data and calculate the gradient,
  3. Take a step downhill in the correct direction based on the gradient,
  4. Step size is a function of the gradient,
  5. Repeat until convergence to a single point (until a tolerated value is reached).
- Second-order derivative methods (Hessian-based methods) are generally more accurate and converge in fewer steps, but they are more resource intensive. In contrast, first-order derivative methods (gradient-

[*] Department of Epidemiology and Biostatistics, College for Public Health and Social Justice, Saint Louis University. Email miao.cai@slu.edu

based methods) are less resource intensive but less accurate. If calculating the Hessian is easy and cheap, we should take the Hessian-based methods.

- The field typically used Newtonian optimization method on unregularized GLM models since it is one of the fastest converging method and it has nice matrix form properties. When the number of parameters are huge or the data have very steep or twisted curvature, the Hessian-based methods will become challenging since they are too expensive.
- This is because deep learning models are usually engaged with huge amounts of data, parameters, and hyperparameters. Typically gradient-based methods are not easy to implement (we have to turn to mini-batch gradient descent or stochastic gradient descent), let alone the Hessian-based methods.

2. Machine learning versus inferential methods (like GLMs): Write a multiple paragraph discussion about ML versus inferential model like GLMs. Address all the points listed below:

(a) What is the overall goal of a machine learning prediction model?

(b) Discuss the concept of training error versus generalization error, and how these two types of error are estimated in practice.

(c) What different forms of cross validation can be implemented? Name a few different schemes for cross validation. Discuss their strengths and weaknesses.

(d) Define model capacity. Is more capacity always better? Why or why not? How much capacity is optimal?

(e) Define regularization – what are different ways we can regularize models in machine learning?

(f) What is the difference between hyperparameters and model parameters? How do we optimize model parameters? How do we optimize model hyperparameters? If we implement an elastic net regression, what are the model parameters, and what are the model hyper parameters?

(g) A clinic manager would like you to create a patient-no show risk model. The goal is to use the no-show risk to schedule extra patients...if a patient is at a high risk to no-show, we will book extra appointments that day, assuming some patients will no-show for their appointment. Should you use a machine learning approach here, or should you use classic inferential models like a GLM? Why? Justify your response.

(h) A physician would like to understand whether drug A or drug B is more effective at controlling blood pressure. We have some data from last year where some patients were exposed to drug A, and some patients were exposed to drug B. Should you implement a machine learning approach to predict blood pressure here (like a GBM), or should you implement inferential models like GLM? Why? Justify your response.

- The overall goal of machine learning is to build a prediction model that has high generalizability to data in the future. Since we can't see data in the future, so usually we want to achieve low training error and low test error.
- Training error is how well the machine learning model is fitted on the training data, while test error is how well the machine learning model based on the training data is fitted onto the test data, or more accurately, the expected value of the error on a new input. In practice, training and testing error are estimated by calculating the mean square error (the difference between predicted results and true results).
- Here are some different forms of cross-validation:
  1. k-fold cross-validation: a partition of the dataset is formed by splitting it into k non-overlapping subsets
  2. leave-p-out cross-validation: using p observations as the validation set and the remaining observations as the training set. This is repeated on all ways to cut the original sample on a validation set of p observations and a training set.

k-fold cross-validation is a non-exhaustive cross-validation and it will not compute all ways of splitting the original sample, which is an approximation of leave-p-out cross-validation. Therefore, k-fold cross-validation is less computationally intensive but less accurate, while leave-p-out cross-validation is more accurate but more computationally intensive.

- Model capacity is its ability to fit a wide variety of functions. Models with low capacity may struggle to fit the training set. Models with high capacity can overfit by memorizing properties of the training set that do not serve them well on the test set. Model capacity should depend on training and testing error. An optimal model capacity should make training and testing error similarly high.
- Regularization is any modification we make to a learning algorithm that is intended to reduce its generalization error but not its training error. There are two types of regularization techniques: L1 method and L2 method.

  L1 method is also known as LASSO, with the following penalty term adding to the loss function: $\lambda \sum_{j=1}^{p} |\beta_i|$. In contrast, L2 method is known as Ridge regression and a different penalty term is added to the loss function, $\lambda \sum_{j=1}^{p} \beta_i^2$.
- Hyperparameters are used to control the behaviors of machine learning models, and their values are not adapted by the learning algorithm itself. In contrast, model parameters are adapted by the learning algorithm. We optimize model parameters by optimizing the loss function of the training set, while we optimize model hyperparameters by using a separate validation set. The model parameters in an elastic network are the βs associated with the features, while hyperparameter ranged between 0 and 1, which controls how much L1 or L2 regularization is used.
- I would prefer a machine learning model in this case since a machine learning model usually has a better prediction accuracy than traditional inferential models. In this case, the manager is not interested in specific predictor variables or how much that predictor is associated with no-risk model.
- I would prefer a traditional inferential model since the physician is specifically interested in the effect of a drug. An inferential model is better at explaining the association between two variables since the theories are simpler. It may also be easier to explain to physicians since they are most likely more familiar with traditional inferential models (GLMs).

3. Online learning/SGD
(a) What is online learning?
(b) How does implementing online learning in sk-learn allow us to fit models on data that is larger than memory

- Online learning or stochastic gradient methods use only a single row at a time to estimate the gradient and optimize the machine learning model.
- The gradient of the loss is estimated each sample at a time and the model is updated along the way with a decreasing strength schedule (aka learning rate).

4. Scikit-learn, Spark, Dask, and h2o: Based on your readings and experience in this class thus far, what do you think the strengths and weaknesses are of using these frameworks for data management and machine learning?
(a) For each technology, address its capabilities for working with data larger than memory, as well as if it can fit models in parallel (using multiple cores or clusters).
(b) Compare and contrast the available models for each framework. Can you fit all the same types of models in each framework?
(c) Compare and contrast the available optimization methods in each framework.

- Working with data larger than memory:

- scikit-learn: scikit-learn is fine for large data sets, but not as good as h2o and dask.
  - spark: good at data management, not for machine learning with big data sets since it will spill excessive data onto disk, which will be much slower than in-memory methods.
  - dask: very good at data larger than memory by using parallel scores or clusters.
  - h2o: very good at dealing with data larger than memory.
- No, we can't fit the same types of models in each framework, here are the available models in each framework:
  - scikit-learn: classification, regression, clustering, dimentionality reduction, model selection, and preprocessing. See scikit-learn official document.
  - spark: classification, regression, linear methods, decision trees, tree ensembles. See Spark official documentation.
  - dask: generalized linear models, clustering, XGBoost, and other regression metrics. See dask-ml documentation.
  - h2o: supervised learning, XGBoost, unsupervised learning, generic models and others. See H2O official documentation
- The available optimization methods in the four frameworks:
  - scikit-learn: linear regression and Ridge use closed-form solution or stochastic gradient descent, LASSO and elastic net use coordinate descent, ARD regression and Bayesian Ridge use something like expectation-maximization algorithm, and HuberRegressor uses BFGS. (referred from What optimization algorithms are used in scikit-learn? From StackExchange)
  - spark: data serialization, memory tuning, memory management, determining memory consumption, and garbage collection tuning,
  - dask: ADMM (Alternating Direction Method of Multipliers), gradient descent, L-BFGS, Newton's method (from dask-ml document)
  - h2o: Generalized ADMM Solver, L-BFGS, OLS, stochastic gradient descent (from this presentation: High Performance Machine Learning in R with H2O)

5. Deep learning: Write a few paragraphs discussing deep learning that address the points below.

(a) How does deep learning generally differ from other types of machine learning, like Elastic net, random forest, or GBM? How is it similar?

(b) Why has deep learning received so much attention in the past 8 years? What conditions in data science have contributed to this focus on deep learning?

(c) Do you think deep learning will reshape data science in healthcare in the coming years? Why or why not? What tasks will deep learning succeed or fail at?

- Deep learning is a particular kind of machine learning that achieves great power and flexibility by learning to represent the world as a nested hierarchy of concepts, with each concept defined in relation to simpler concepts, and more abstract representations computed in terms of less abstract ones. Traditional machine learning methods such as elastic net and random forest need to be told how to make an accurate prediction by tuning hyperparameters, regularization. In contrast, deep learning is able to learn that through its own data processing. The similarity between machine learning and deep learning is that they both use complex models and optimization to try to make better predictions.

- The fast growth of huge datasets, the increase of computational power and GPU contributed to the increase of attention in deep learning in the past eight years.

- I believe that deep learning will change data science in healthcare in the coming years. In the future, we would have better quality and huge amounts of data, more powerful computers, and better optimization algorithms to enable deep learning to make more accurate predictions. Deep learning will succeed in aspects that follow strong or similar patterns, such as diagnosis through images, but less successful in aspects that are less predictable, such as human emotion.