Optimizing Machine Learning Model Memory Usage With Minimal Statistical Performance Degradation

Authors: Saharat Rodjanamongkol and Pakin Wirojwatanakul

The performance of a machine learning model can be broken down into statistical performance and hardware performance. In practice, oftentimes improving the statistical performance comes at the expense of a degradation in hardware performance. For example, creating new features via the process of feature engineering can improve the statistical performance, but degrade the hardware performance because the new transformed data will have higher memory requirements.

In this project we seek to explore various dimension reduction methods to determine which methods can maximally conserve memory (optimizing hardware performance) while minimally degrading statistical performance of the models. Let the original data have d dimensions and the reduced data have m dimensions where $m << d$. By performing the classification in the reduced feature space instead of the original feature space, the memory cost of storing a data point can be reduced from $O(d)$ to $O(m)$. Minimizing memory usage for a machine learning model can lead to improvements in the financial bottom line of companies deploying the model because memory is expensive, especially when the data set is large.

For the investigation, we will compare the errors (number of misclassified points) from data reduced by various methods such as (PCA, Kernel PCA, Random Projection, Factor Analysis, ISOMAP, etc) against each other and the error of the classifier in the original space. For each dimensionality reduction algorithm, we will also compare how each performs as $m \to 0$ and find the m that best compromises between statistical and hardware performance. Since, the focus of this investigation is on the dimension reduction methods, we will use Random Forest as the classifier performs well out of the box and doesn't require much parameter tuning.

We plan to compare around 20 different classification datasets from https://www.openml.org/search?type=data and https://archive.ics.uci.edu/ml/index.php in order to see how well different algorithms work on different dataset and to see if there're any underlying trends on the algorithms' effectiveness. Our main criteria for the dataset are that they're classification dataset with at least 10 features and 500 entries so that there are dimensions to reduce and enough entries to train the algorithm. One example of a dataset we really like is the wine quality dataset https://archive.ics.uci.edu/ml/datasets/Wine+Quality . It is a classification dataset with a sufficient number of features and entries, 12 features and 4898 entries. The values are also continuous so some algorithm would be able to learn the underlying structure or manifold (space that is locally euclidean) of the data well. Most importantly, the data is interpretable, objective and has less noise. The features are different chemical properties of the wine taken from a physicochemical test. We hope to be able to explain why the algorithm makes the reduction.