CS772 Project Proposal

Mihir, Praneat, Rahul, Ravija, Sanat

February 2024

1 Prologue

Training on web-scraped data has revolutionised machine learning, but it comes with its own set of problems. The sheer magnitude of data causes extremely long training times, but much of the time is wasted on either learning redundant data-points or on attempting to learn points that are not learnable. In this project, we explore methods to speed up training by smartly selecting points to train on.

2 Seed Paper

Our seed paper is Prioritized Training on Points that are learnable, Worth Learning, and Not Yet Learnt. The paper explores functions that can *rank* data points to massively reduce the training time without suffering any hits on the accuracy. It starts with explaining the two prevalent methods for prioritizing samples - filtering high noise examples, and training greedily on high loss examples - and their drawbacks. The authors go on to introduce a custom metric called the **Reducible Holdout Loss Selection** for selecting points to train on.

On the large web-scraped image dataset Clothing-1M, RHO-LOSS trains in 18x fewer steps and reaches 2% higher accuracy.

3 Possible Paths

3.1 Verification and Adaptation

The authors only primarily show Computer Vision benchmarks, where it is relatively easier to quantify and eliminate noise (simple autoencoders have been seen to perform remarkably well), we can look at **adapting** the model for tasks in NLP and Signal Processing and testing them on benchmark datasets. This might be computationally expensive, so this would also require some smart batching and load distribution strategies.

3.2 Learnable Metrics

Auto-encoders have been used to filter out un-learnable points in many domains, especially signal processing. We can explore if probabilistic auto-encoder architectures can replace **RHO Loss** and can possibly have more interesting applications. Using different auto-encoder architectures is akin to learning complex metrics for ranking training samples.

3.3 Tweaking Loss Function

While the **RHO Loss** is very intuitive and easily computable, we feel training on a small held out set is not enough for making any solid comments, especially if subsets are expected to have wildly varying distributions. We can also look at modifications to the loss function and test it on an artificial dataset.

4 Expected Deliverables

1. Benchmarks on standard NLP and SigProc datasets.
2. Alternative architectures, and their benchmarks.