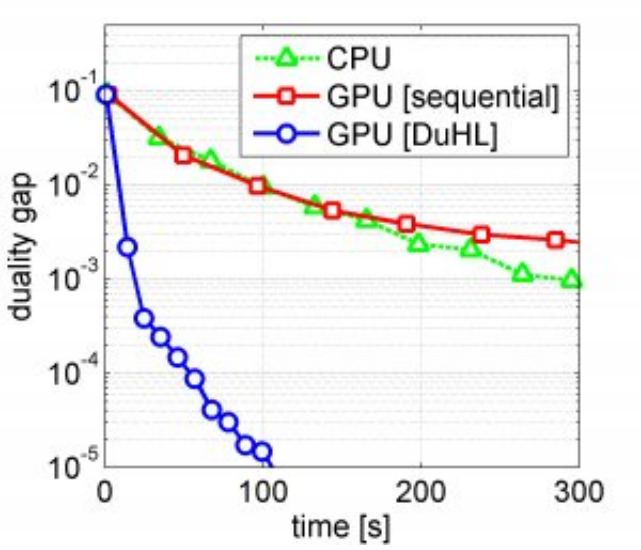
IBM scientists demonstrate 10x faster large-scale machine learning using GPUs

December 5 2017



DuHL in action for the application of training large-scale Support Vector Machines on an extended, 30GB version of the ImageNet database. Credit: IBM Blog Research

Together with EPFL scientists, our IBM Research team has developed a scheme for training big data sets quickly. It can process a 30 Gigabyte training dataset in less than one minute using a single graphics processing unit (GPU)—a 10x speedup over existing methods for limited memory training. The results, which efficiently utilize the full potential of the GPU, are being presented at the 2017 NIPS Conference in Long Beach, California.

Training a machine learning model on a terabyte-scale dataset is a common, difficult problem. If you're lucky, you may have a server with enough memory to fit all of the data, but the training will still take a very long time. This may be a matter of a few hours, a few days or even weeks.

Specialized hardware devices such as GPUs have been gaining traction in many fields for accelerating compute-intensive workloads, but it's difficult to extend this to very data-intensive workloads.

In order to take advantage of the massive compute power of GPUs, we need to store the data inside the GPU memory in order to access and process it. However, GPUs have a limited memory capacity (currently up to 16GB) so this is not practical for very large data.

One straightforward solution to this problem is to sequentially process the data on the GPU in batches. That is, we partition the data into 16GB chunks and load these chunks into the GPU memory sequentially.

Unfortunately, it is expensive to move data to and from the GPU and the time it takes to transfer each batch from the CPU to the GPU can become a significant overhead. In fact, this overhead is so severe that it may completely outweigh the benefit of using a GPU in the first place.

Our team set out to create a technique that determines which smaller part of the data is most important to the training algorithm at any given time. For most datasets of interest, the importance of each data-point to the training algorithm is highly non-uniform, and also changes during the training process. By processing the data-points in the right order we can learn our model more quickly.

For example, imagine the algorithm was being trained to distinguish between photos of cats and dogs. Once the algorithm can distinguish that

a cat's ears are typically smaller than a dog's, it retains this information and skips reviewing this feature, eventually becoming faster and faster.

This is why the variability of the data set is so critical, because each must reveal additional features that are not yet reflected in our model for it to learn. If a child only looks outside and the sky is always blue, they will never learn that it gets dark at night or that clouds create shades of gray. It's the same here.

This is achieved by deriving novel theoretical insights on how much information individual training samples can contribute to the progress of the learning algorithm. This measure heavily relies on the concept of the duality gap certificates and adapts on-the-fly to the current state of the training algorithm, i.e., the importance of each data point changes as the algorithm progresses. For more details about the theoretical background, see our current paper.

Taking this theory and putting it into practice we have developed a new, re-useable component for training machine learning models on heterogeneous compute platforms. We call it DuHL for Duality-gap based Heterogeneous Learning. In addition to an application involving GPUs, the scheme can be applied to other limited memory accelerators (e.g. systems that use FPGAs instead of GPUs) and has many applications, including large data sets from social media and online marketing, which can be used to predict which ads to show users. Additional applications include finding patterns in telecom data and for fraud detection.

We show DuHL in action for the application of training large-scale Support Vector Machines on an extended, 30GB version of the ImageNet database. For these experiments, we used an NVIDIA Quadro M4000 GPU with 8GB of memory. We can see that the scheme that uses sequential batching actually performs worse than the CPU alone,

whereas the new approach using DuHL achieves a 10x speed-up over the CPU.

The next goal for this work is to offer DuHL as a service in the cloud. In a cloud environment, resources such as GPUs are typically billed on an hourly basis. Therefore, if one can train a machine learning model in one hour rather than 10 hours, this translates directly into a very large cost saving. We expect this to be of significant value to researchers, developers and data scientists who needs to train large-scale machine learning models.

This research is part of an IBM Research effort to to develop distributed deep learning (DDL) software and algorithms that automate and optimize the parallelization of large and complex computing tasks across hundreds of GPU accelerators attached to dozens of servers.

Provided by IBM