note of 'Runtime Data Layout Scheduling for Machine Learning Dataset'

zxp

November 30, 2024

1 content

This is a paper from 2017. The core of this paper is the impact of different data layouts on machine learning. It's not really related to what we did here, but the paper proposes that choosing the right data has a big impact on the speed of training. But the paper refers to the data layout suitable for machine learning,DEN (Dense), CSR (Compressed Sparse Row),ELL (ELLPACK/ITPACK), COO (Coordinate),and DIA (Diagonal). CSR is usually used on CPU and DEN is used on GPU. This paper compares the performance of these data layouts in different cases. The paper's work on deep learning is to compare the effect of different hardware and batch on training speed.

1.1 BACKGROUND

The paper points out that "typical ML applications are usually data-intensive, memory-constrained and irregularly accessed, so their huge parallelism and complex memory hierarchy form an obstacle to efficient parallel ML design. Data formats have a significant impact on storage and computation complexity, memory bandwidth, and the efficiency of parallel processing. However, for dnn, the paper points out that each iteration can only process a small part of the data due to the limitation of the algorithm, and multiple iterations are needed, but the paper only looks for suitable parameters to better iterate.

1.2 MEMORY-EFFICIENT SVM

The main focus of this paper is the binary classification machine learning method, SVM. The kernel matrix of SVM is too large, and decomposition and iterative methods have been devised to deal with SVM kernels. SMO (SequentialMinimal Optimization) based on decomposition method is used to calculate element position.

In many practical scientific computing applications, the matrix is still sparse. For extremely sparse datasets, DEN must store M·N elements, while COO and CSR only need to store elements with the number of nonzero elements, and block variants are often used when many dense sub-blocks exist in the sparse matrix. The computational complexity of SVM is proportional to the storage complexity. Existing SVM tools (such as LIBSVM and GPUSVM) use a fixed data format for all datasets. However, the paper's experiments show that for different datasets, the most appropriate format varies significantly, with the best format being 3.73 to 14.3 times faster than the worst format for the same dataset. The performance of SVMS is also limited by memory bandwidth, which varies when the same dataset is processed using different formats. The paper analyzes the different data layouts and extracts a few important parameters to help choose the right data layout, as shown in the chart of the paper (in fact, it is almost useless for DNNS, most of it is related to the density of the data matrix and the distribution of non-zero elements, but sparse matrices are unheard of for DNNS).

1.3 DEEP LEARNING

The paper states that a fast implementation of deep learning requires: choosing the right hardware, adjusting the batch size (512 is chosen in this paper because 512 is the fastest on the hardware and the size is the power of 2 n̂), adjusting the learning rate, and adjusting the momentum. The paper does not give theoretical guidance to explain how to optimize, but in a range of constantly try, and because of the long time, the selected data set is not large, hardware performance and now there is a huge gap, basically no reference.

1.4 Experimental

For SVM, the paper flexible selection data layout is faster than the fixed data layout. For deep learning, papers tune parameters faster than not

2 Feelings

The core of this paper is to adjust the different data layout of SVM and point out that the speed of machine learning is greatly affected by the data layout. It has nothing to do with us