cuTimeWarp: Accelerating Soft Dynamic Time Warping on GPU

Alex Kyllo Afrooz Rahmati

March 4, 2021

Abstract

This report explores techniques for optimizing the computation of Soft Dynamic Time Warping, a differentiable sequence dissimilarity measure, on graphics processing units (GPU), for the purpose of enabling high-performance machine learning on time series datasets.

Introduction

Time series machine learning is a research area with countless useful applications such as recognizing sounds and gestures. Clustering or classifying large time series datasets is challenging partly because of the need to define a measure of dissimilarity between two time series. Practical applications require finding common structure despite different speeds or phases; a word means the same whether spoken quickly or slowly. Another requirement is that the measure must be differentiable so that its gradient can be used as a loss function to minimize in model fitting. Finally, the measure must be efficient to calculate. To this end we will explore GPU acceleration of Soft Dynamic Time Warping (Soft-DTW) [1], a differentiable dissimilarity measure, to enable high performance time series machine learning.

Dynamic Time Warping (DTW) is an algorithm to compute the dissimilarity between two time series, which may vary in speed and phase. The basic algorithm for DTW is to use Bellman's recursion, a dynamic programming technique, to find the lowest-cost path diagonally across a pairwise distance matrix. The computation cost for this approach is quadratic(mn) for time series vectors of length m and n [1]. The formula for the DTW between time series x and y is:

$$DTW(x,y) = min_{\pi}\sqrt{\sum_{(i,j)\in\pi} d(x_i,y_j)^2}$$

Where $d(x_i, y_j)^2$ is the cost function, typically pairwise squared Euclidean distance. The loss function for DTW is not differentiable due to the min operation within the formula; a small change in the input time series may result in zero change in the path cost. However, we can create a differentiable version called Soft-DTW by replacing the min with a soft-min function [1]:

$$\text{soft-min}_{\gamma}(a_1,...,a_n) = -\gamma log\sum_i e^{-a_i/\gamma}$$

Hence, Soft-DTW is parameterized by the smoothing constant gamma, which becomes a tunable hyperparameter in machine learning model training applications.

A common technique in machine learning with Soft-DTW is the computation of barycenters, which are centroids within the space of a set of time series. The differentiability of Soft-DTW allows for barycenter finding via gradient descent, and then new observations can be clustered or classified by finding the closest barycenter.

Related Work

One important optimization we can apply is the use of Sakoe-Chiba bands, which prune the warping path search space, allowing path finding in subquadratic time. While this technique produces an approximation of the optimal path, in practice it has been shown to improve task performance by preventing pathological alignments where a very small portion of one time series maps onto a large portion of another [2].

Utilizing indexing to construct lower bounds on warping distance is an optimization technique for speeding up nearest neighbor search via early removal of poor candidates [2]. Shen and Chi (2021) proposes an optimization of nearest neighbor search of multivariate time series, leveraging the triangle inequality and quantization-based point clustering to restrict the search [3].

Xiao et al (2013) introduced a prefix computation technique for transforming the diagonal data dependence to improve parallel computation of the cost matrix on GPU [4]. Zhu et al (2018) demonstrates a method of optimizing memory access by taking advantage of the diagonal data dependency to rearrange the matrix so that elements on the same diagonal are stored contiguously [5]. A prior implementation of Soft-DTW on CUDA using PyTorch and Numba is capable of 100x performance improvement over the original Soft-DTW Cython code, but is limited to sequence lengths of 1024 (CUDA max block size) and leaves many opportunities for further CUDA optimizations such as the use of shared memory [6]. In our project we will focus on this area of opportunity, optimizing matrix structure and memory access patterns to maximize parallelism and minimize memory latency in the computation of the warping path matrix.

Methods

Results

Discussion

References

[1] M. Cuturi and M. Blondel, "Soft-DTW: A differentiable loss function for time-series," *arXiv:1703.01541 [stat]*, Feb. 20, 2018. arXiv: `1703.01541`. [Online]. Available: `http://arxiv.org/abs/1703.01541` (visited on 01/16/2021).

[2] E. Keogh, "Exact indexing of dynamic time warping," in *Proceedings of the 28th international conference on Very Large Data Bases*, ser. VLDB '02, Hong Kong, China: VLDB Endowment, Aug. 20, 2002, pp. 406–417. (visited on 02/14/2021).

[3] D. Shen and M. Chi, "TC-DTW: Accelerating multivariate dynamic time warping through triangle inequality and point clustering," *arXiv:2101.07731 [cs]*, Jan. 19, 2021. arXiv: `2101.07731`. [Online]. Available: `http://arxiv.org/abs/2101.07731` (visited on 02/14/2021).

[4] L. Xiao, Y. Zheng, W. Tang, G. Yao, and L. Ruan, "Parallelizing dynamic time warping algorithm using prefix computations on GPU," in *2013 IEEE 10th International Conference on High Performance Computing and Communications 2013 IEEE International Conference on Embedded and Ubiquitous Computing*, Nov. 2013, pp. 294–299. DOI: `10.1109/HPCC.and.EUC.2013.50`.

[5] H. Zhu, Z. Gu, H. Zhao, K. Chen, C. Li, and L. He, "Developing a pattern discovery method in time series data and its GPU acceleration," *Big Data Mining and Analytics*, vol. 1, no. 4, pp. 266–283, Dec. 2018, Conference Name: Big Data Mining and Analytics, ISSN: 2096-0654. DOI: `10.26599/BDMA.2018.9020021`.

[6] M. Maghoumi, *Maghoumi/pytorch-softdtw-cuda*, original-date: 2020-05-02T23:28:24Z, Jan. 21, 2021. [Online]. Available: `https://github.com/Maghoumi/pytorch-softdtw-cuda` (visited on 01/23/2021).