Learning Fast Algorithms for Linear Transforms Using Butterfly Factorizations

Tri Dao1, Albert Gu1, Matthew Eichhorn1, Atri Rudra1, Christopher Ré1

triad@stanford.edu, albertgu@stanford.edu, maeichho@buffalo.edu, atrir@buffalo.edu, christiner@cstanford.edu

Department of Computer Science, † Stanford University, ‡ University at Buffalo, SUNY


Overview

Fast linear transforms are important and ubiquitous in ML.
- Examples: discrete Fourier transform (DFT), discrete cosine transform (DCT), Hadamard transform, convolution.
- Uses: preprocessing, random projection, neural net compression.
- Compared to hard-coded algorithms, learned algorithms offer:
  - easy integration with ML frameworks (PyTorch/TF).
  - flexible adaptation to tasks/datasets.

Compared to hard-coded algorithms, learned algorithms offer:
- expressivity: Butterfly factorizations and their products (BPBP) can capture:
  - classic discrete transforms: DFT, DCT, DST, Hadamard, convolution.
  - iterative: Fastfood [Le et al., 2013], orthogonal random features [Yu et al., 2016].
  - structured matrices in deep learning: low-rank, Toeplitz-like, ACDC [Moczulski et al., 2016].

Recursive matrix factorization

FFT case study:

|------|------|------|------|------|------|------|------|

N/2-point DFT

Butterfly matrix for N = 16.

3 binary choices for each permutation P_k.

Butterfly Permutation (BP):

\[
B_N = \begin{bmatrix}
B_{N/2} & \cdots & 0 \\
0 & \cdots & B_{N/2} \\
\end{bmatrix}
\]

Butterfly parameterization recovers fast algorithms for common transforms up to \( N = 1024 \) and convolutions up to \( N = 512 \).

Efficiency

Theorem 1. An \( N \times N \) matrix in our proposed real/complex butterfly class has \( 4N \) or \( 8N \) parameters respectively, and can be multiplied by a vector with \( O(N \log N) \) computations.

Butterfly matrix: \( 4N \) parameters, \( O(N \log N) \) algorithm.

Learning Fast Algorithms for Discrete Transforms

Butterfly matrix: \( 4N \) parameters, \( O(N \log N) \) algorithm.

Theorem 2. The VC dimension of neural networks with \( L \) butterfly weight layers and \( W \) total parameters is \( O(LW \log W) \).

Sample complexity: almost linear in number of params.

Compact Neural Networks

MobileNet (with Butterfly 1x1 Conv) on CIFAR-10.

Model | Num. params | Test accuracy |
--- | --- | --- |
MobileNet | 3.2M | 91.6% |
2.0 MobileNet w/ Butterfly 1x1 Conv | 2.5M | 92.9% |
0.5 MobileNet | 0.8M | 89.7% |
MobileNet w/ Butterfly 1x1 Conv | 0.7M | 91.0% |

Higher accuracy with fewer parameters, even on architecture already designed for efficiency.