Band-limited Training and Inference for Convolutional Neural Networks

Adam Dziedzic*, John Paparrizos*, Sanjay Krishnan, Aaron Elmore, Michael Franklin
Natural images
More information put in lower frequencies

Original image

Spatial domain
Frequency domain
Natural images
Transformations between the domains

Compression 50%

Spatial domain → Frequency domain

FFT → IFFT
Method for ConvNets to constrain the frequency band in convolution operation for efficiency

Compression 50% in practice
FFT based convolution

Mathieu et al.: "Fast Training of Convolutional Networks through FFTs"
Vasilache et al.: "Fast Convolutional Nets With fbfft: A GPU Performance Evaluation"
FFT based convolution

Mathieu et al.: “Fast Training of Convolutional Networks through FFTs”
FFT based convolution

Mathieu et al.: “Fast Training of Convolutional Networks through FFTs”
FFT based convolution

Mathieu et al.: “Fast Training of Convolutional Networks through FFTs”
FFT based convolution

Mathieu et al.: “Fast Training of Convolutional Networks through FFTs”
cuDNN: Substantial memory workspace needed for intermediate results.
Band-limited FFT based convolution

Band-liming = masking out high frequencies

\[ \text{Data: } x \xrightarrow{\text{FFT}} \text{xfft} \xrightarrow{\text{Band-limited}} \text{ Band-limited (xfft)} \xrightarrow{\text{xCfft}} \text{offt} \xrightarrow{\text{IFFT}} \text{Out: } o \]

\[ \text{Filter: } y \xrightarrow{\text{FFT}} \text{yfft} \xrightarrow{\text{Band-limited}} \text{ Band-limited (yfft)} \xrightarrow{\text{yCfft}} \text{offt} \xrightarrow{\text{IFFT}} \text{Out: } o \]

Band-liming = masking out high frequencies
Band-limited FFT based convolution

Data: $x$

FFT($x$) → $x^{\text{fft}}$ → Band-limited ($x^{\text{fft}}$) → $x^{\text{Cfft}}$ → Less memory used → IFFT($o^{\text{fft}}$) → Out: $o$

Filter: $y$

FFT($y$) → $y^{\text{fft}}$ → Band-limited ($y^{\text{fft}}$) → $y^{\text{Cfft}}$ → $x^{\text{Cfft}}$ → $x^{\text{fft}}$ → $y^{\text{fft}}$ → $y^{\text{Cfft}}$
Band-limited FFT based convolution

Data: $x$

Filter: $y$

FFT($x$) -> xfft

FFT($y$) -> yfft

Band-limited (xfft)

Band-limited (yfft)

xCfft $\bullet$ yCfft

IFFT(offt)

Out: $o$

Less memory used

Faster computation
Band-limited FFT based convolution

Preserve enough of the spectrum to retain high accuracy of models.

Data: x

FFT(x) \rightarrow xfft

Band-limited (xfft)

Band-limited (yfft)

Filter: y

FFT(y) \rightarrow yfft

xCfft \bigcirc yCfft

Less memory used

Faster computation

IFFT(offt) \rightarrow Out: o
Band-limiting Technique

1. FFT of an input data
Band-limiting Technique

1. FFT of an input data
2. Conjugate symmetry

\[
\begin{bmatrix}
1 & -j & 1+j
\end{bmatrix}
\]
Band-limiting Technique

1. FFT of an input data
2. Conjugate symmetry
Band-limiting Technique

1. FFT of an input data
2. Conjugate symmetry
3. Real values
Band-limiting Technique

1. FFT of an input data
2. Conjugate symmetry
3. Real values
4. No constraints
1. FFT of an input data
2. Conjugate symmetry
3. Real values
4. No constraints
5. 1st compression
Band-limiting Technique

1. FFT of an input data
2. Conjugate symmetry
3. Real values
4. No constraints
5. 1st compression
Band-limiting Technique

1. FFT of an input data
2. Conjugate symmetry
3. Real values
4. No constraints
5. 1\textsuperscript{st} compression
6. 2\textsuperscript{nd} compression
Band-limiting Technique

1. FFT of an input data
2. Conjugate symmetry
3. Real values
4. No constraints
5. 1\textsuperscript{st} compression
6. 2\textsuperscript{nd} compression
Effects of band-limiting on accuracy

ResNet-18 on CIFAR-10
Effects of band-limiting on accuracy

ResNet-18 on CIFAR-10

Test Accuracy (%) vs. Compression rate (%)
Effects of band-limiting on accuracy

Test Accuracy (%) vs. Compression rate (%)

ResNet-18 on CIFAR-10
Effects of band-limiting on accuracy

Test Accuracy (%) vs Compression rate (%)

ResNet-18 on CIFAR-10

- 93.5%
- 92%

DenseNet-121 on CIFAR-100

- 75.3%
- 71.2%
Main **take-aways** from Band-limited CNNs

- Method to constrain the frequency band in convolution.
- Models trained with band-limiting **gracefully degrade** the accuracy as the function of the compression rate.
- Effectively **control resource usage** (GPU/CPU and memory).
- The **low frequency** coefficients learned first during training.
- The **same compression rate** applied to training and inference.
- The more band-limited model, the more **robust to attacks**.
- Applicable to **other domains**: time-series & speech data.
Thank you

Poster: 6:30-9:00 PM @ Pacific Ballroom #132

github.com/adam-dziedzic/bandlimited-cnn

ady@uchicago.edu
Backup
Why is FFT based convolution important?

- The theoretical properties of the Fourier domain are well understood. No such properties in other domains (Winograd).
- ResNet and DenseNet architectures use 7x7 filters in first layers.
- FFT based convolution can be combined with spectral pooling.
- Band-limiting in the first layer serves as a simple defense.
- A standard algorithm included in popular frameworks (cuDNN).
- Gradient acts as a large filter in the backward pass.
- Zlateski et al. suggest using FFT based convolution on CPUs.
- The 1D FFT convolution for DSP where large filters are used.
Band-limited FFT based convolution formally

Cross-correlate input data and filter: \( x \ast_c y \)

\[
F_x[\omega] = F(x[n]) \quad F_y[\omega] = F(y[n])
\]

\[
x \ast_c y = F^{-1}(F_x[\omega] \odot F_y[\omega])
\]

Spectrum of convolution: \( S[\omega] = F_x[\omega] \odot F_y[\omega] \)

\[
M_c[\omega] = \begin{cases} 
1, & \omega \leq c \\
0, & \omega > c 
\end{cases}
\]

\[
x \ast_c y = F^{-1}[(F_x[\omega] \odot M_c[\omega]) \odot (F_y[\omega] \odot M_c[\omega])] \\
x \ast_c y = F^{-1}(S[\omega] \odot M_c[\omega])
\]

Energy (Parseval's theorem):

\[
\sum_{n=0}^{N-1} |x[n]|^2 = \sum_{\omega=0}^{2\pi} |F_x(\omega)|^2
\]
Robustness to noise

![Graph showing robustness to noise with different levels of Gaussian noise. The graph plots test accuracy (%) against the level of Gaussian noise (sigma). Different lines represent different conditions: FP32-C=0% full spectra, FP16-C=0% full spectra (reduced precision: 16 bits), FP32-C=0% early stopping, FP32-C=50% band-limited, and FP32-C=85% band-limited.]
Compression Rate for Training vs Inference

DenseNet-121 on CIFAR-100

Test accuracy (%) vs Inference Compression Rate (%)

Train compression: C=50, C=75
Compression Rate for Training vs Inference

DenseNet-121 on CIFAR-100

Test accuracy (%) vs Inference Compression Rate (%)

Train compression: C=0, C=50, C=75, C=85
Effectively control resource usage

ResNet-18 on CIFAR-10

Normalized performance (%)

- GPU memory allocated
- Epoch time

Compression rate (%)
Compression Rate for Training vs Inference

Test accuracy (%) vs Inference Compression Rate (%)

ResNet-18 on CIFAR-10

Train Compression Rate (%):

- C=0
Compresssion Rate for Training vs Inference

- **Test accuracy (%)**
- **Inference Compression Rate (%)**

*ResNet-18 on CIFAR-10*

**Train Compression Rate (%)**:
- C=0
- C=85
Compression Rate for Training vs Inference

Test accuracy (%) vs Inference Compression Rate (%)

- **ResNet-18 on CIFAR-10**

Smooth degradation of accuracy during inference

**Train Compression Rate (%)**:
- $C=0$
- $C=85$
Compression Rate for Training vs Inference

ResNet-18 on CIFAR-10

Apply the same compression rate to training and inference.
Tuning: Accuracy vs Higher Performance

**ResNet-18 on CIFAR-10**

Test Accuracy (%)

- 95
- 90
- 85
- 80
- 75
- 70
- 65
- 60
- 55
- 50
- 45
- 40
- 35
- 30
- 25
- 20
- 15
- 10
- 5
- 0

Compression rate (%)

- 0
- 20
- 40
- 60
- 80

**DenseNet-121 on CIFAR-100**

Test Accuracy (%)

- 80
- 75
- 70
- 65
- 60
- 55
- 50
- 45
- 40
- 35
- 30
- 25
- 20
- 15
- 10
- 5
- 0

Compression rate (%)

- 0
- 20
- 40
- 60
- 80

**GPU memory allocated**

- 100
- 50
- 0

**Epoch time**

- 100
- 50
- 0

- 0
- 20
- 40
- 60
- 80
“Speaking of longer term, it would be nice if the community migrated to a fully open sourced implementation for all of this [convolution operations, etc.]. This stuff is just too important to the progress of the field for it to be locked away in proprietary implementations. The more people working together on this the better for everyone. There's plenty of room to compete on the hardware implementation side.”

Scott Gray

https://github.com/soumith/convnet-benchmarks/issues/93