高速フーリエ変換で畳み込み
フーリエ変換(FFT)入門

tmtk

Topics

こんにちは。データサイエンスチームのtmtkです。
前回の記事（1.離散フーリエ変換入門）では、離散フーリエ変換を紹介しました。今回の記事では、離散フーリエ変換を高速に計算できる高速フーリエ変換（FFT）というアルゴリズムを紹介します。

高速フーリエ変換とは

高速フーリエ変換は、離散フーリエ変換を分割統治法によって高速に計算します。計算量は入力のベクトル$x \in \mathbb{C}^N$の次元$N \in \mathbb{N}$に対して$O(N \log N)$になります。
離散フーリエ変換の定義を思い出すと、$x \in \mathbb{C}^N$の離散フーリエ変換

$$\mathcal{F}(x) = \begin{pmatrix} \mathcal{F}(x)(1) \\ \vdots \\ \mathcal{F}(x)(N) \end{pmatrix} \in \mathbb{C}^N$$

は

$$\mathcal{F}(x)(n) = \frac{1}{\sqrt{N}} \sum_{t=1}^{N} x(t) \exp(-\frac{i2\pi nt}{N})$$

で定義されるのでした。これを素朴に計算すると、$\mathcal{F}(x)(n)$を計算するために$N+1$の掛け算と$N-1$回の足し算が必要なため、$\mathcal{F}(x)(n)$を計算するのに$O(N)$時間かかります。これをnを動かして繰り返すと、$\mathcal{F}(x)$を計算するのに全体として$O(N^2)$の計算量がかかります。高速フーリエ変換では、これが$O(N \log N)$に改善されます。
いま、次元数Nが偶数であると仮定してこれを少し変形すると、

$$\begin{aligned}
\mathcal{F}(x)(n) =& \frac{1}{\sqrt{N}} \sum_{t=1}^{N} x(t) \exp(-\frac{i2\pi nt}{N}) \\
=& \frac{1}{\sqrt{N}} \sum_{t=1}^{N/2} x(2t-1) \exp(-\frac{i2\pi n(2t-1)}{N}) + \frac{1}{\sqrt{N}} \sum_{t=1}^{N/2} x(2t) \exp(-\frac{i2\pi n2t}{N}) \\
=& \frac{\exp(\frac{i2\pi n}{N})}{\sqrt{2}} \frac{1}{\sqrt{N/2}} \sum_{t=1}^{N/2} x(2t-1) \exp(-\frac{i2\pi nt}{N/2}) + \frac{1}{\sqrt{2}} \frac{1}{\sqrt{N/2}} \sum_{t=1}^{N/2} x(2t) \exp(-\frac{i2\pi nt}{N/2})
\end{aligned}$$

と計算できます。$N/2$次元のベクトル$y, z \in \mathbb{C}^{N/2}$を$y(t) = x(2t-1), z(t) = x(2t) \, (t = 1, 2, \ldots, N)$で定めれば、

$$\mathcal{F}(x)(n) = \frac{\exp(\frac{i2\pi n}{N})}{\sqrt{2}} \mathcal{F}(y)(n) + \frac{1}{\sqrt{2}} \mathcal{F}(z)(n)$$

と書くことができます。ただし$\cdot = y, z \in \mathbb{C}^{N/2}$に対して$\mathcal{F}(\cdot)$は$N/2$次元ベクトル空間$\mathbb{C}^{N/2}$の元ですが、$\mathcal{F}(\cdot)(n + N/2) = \mathcal{F}(\cdot)(n)$で$\mathcal{F}(\cdot) \in \mathbb{C}^N$に自然に拡張して同一視します。なお、いまは$N$が偶数の場合について述べましたが、それ以外の場合でも適宜修正すれば同様の計算ができます。

以上の議論により、次のことがわかります。$x \in \mathbb{C}^N$の離散フーリエ変換$\mathcal{F}(x) \in \mathbb{C}^N$を計算するには、$N/2$次元のベクトル$y, z$を$y(t) = x(2t-1), z(t) = x(2t)$で定め、$y, z$の離散フーリエ変換を計算し、その線形結合

$$\mathcal{F}(x)(n) = \frac{\exp(\frac{i2\pi n}{N})}{\sqrt{2}} \mathcal{F}(y)(n) + \frac{1}{\sqrt{2}} \mathcal{F}(z)(n)$$

を計算すればよい。
この方法で繰り返し対象のベクトルの次元を小さくしていき、再帰的に計算して離散フーリエ変換

$\mathcal{F}(x)$を計算するアルゴリズムを、高速フーリエ変換と呼びます。この分割統治法により、離散フーリエ変換が時間計算量$O(N \log N)$で計算できます。
まったく同じ方法で、逆離散フーリエ変換も高速に計算することができます。この方法を逆高速フーリエ変換とよびます。

まとめ

この記事では、高速フーリエ変換（FFT）のアルゴリズムを紹介しました。高速フーリエ変換は離散フーリエ変換を高速に計算するアルゴリズムで、分割統治法を用いて、N次元のベクトルを時間計算量$O(N \log N)$で離散フーリエ変換できます。
次の記事では、高速フーリエ変換の応用として、畳み込み処理を高速化する方法を紹介します。

参考文献

- 山下幸彦他『工学のためのフーリエ解析』

データ分析と機械学習とソフトウェア開発をしています。 アルゴリズムとデータ構造が好きです。

GCPの利用料が安くなる|GCPの請求代行・運用代行・導入移行支援AWSアの

2020.5.18 **GCP**2020.3

About us会社情報
セミナー・イベント
採用情報
執筆者への取材依頼
フォトギャラリー

Category AWS
Data Science
Tech
Event
Column
Tags
Members

商標について 個人情報保護方針 ISMS認証