Prof. Marc Alexa, Ugo Finnendahl,
Dimitrios Bogiokas, Gabriel Dogadov

Wissenschaftliches Rechnen - Übung 5.2

Schnelle Fourier-Transformation

22.01.2024 bis 26.01.2024

Aufgabe 1: Schnelle Fourier-Transformation (FFT)

1. Welche Laufzeit hat die diskrete Fourier-Transformation, wenn man sie naiv (wie auf dem letzten Aufgabenblatt gezeigt) implementiert?

   Lösung

   Die naive Implementierung der DFT hat eine Laufzeit von $\mathcal{O}(n^2)$, selbst wenn man die DFT-Matrix gar nicht aufstellt und die Einträge direkt berechnet.

   Lösung Ende

2. Konstruieren Sie die DFT-Matrix $\mathbf{\Omega}_{2n}$ mithilfe der DFT-Matrix $\mathbf{\Omega}_n$, einer $2n \times 2n$ Permutationsmatrix $\mathbf{P}_{2n}$, sowie einer $n \times n$ Diagonalmatrix $\mathbf{F}_n$. Wie kann man diese Eigenschaft nutzen, um die Fourier-Transformation von Signalen der Länge 2^n effizient zu berechnen?

   Lösung

   Zunächst ist $\mathbf{F}_n$ eine $n \times n$ Diagonalmatrix, die alle Potenzen von ω_{2n} bis zur $(n-1)$-ten Potenz enthält:

   $$\mathbf{F}_n = \operatorname{diag}(1, \omega_{2n}, \omega_{2n}^2, \dots, \omega_{2n}^{n-1}).$$

   Dann ist $\mathbf{P}_{2n}$ eine $2n \times 2n$ Permutationsmatrix, die alle geraden Einträge des Signals nach oben und die ungeraden Einträge nach unten sortiert (aber die Reihenfolge unter ihnen beibehält).

   $$\mathbf{P}_{2n} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \dots & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 & 1 \end{bmatrix} \begin{matrix} \left.\vphantom{\begin{matrix}1\\0\\\vdots\\0\end{matrix}}\right\} \text{gerade Einträge} \\ \left.\vphantom{\begin{matrix}1\\0\\\vdots\\0\end{matrix}}\right\} \text{ungerade Einträge} \end{matrix}$$

   Permutationsmatrizen sind orthogonal, sodass $\mathbf{P}_{2n}^\mathsf{T}$ ihre Inverse ist. Es gilt folgende Korrespondenz:

   $$\mathbf{\Omega}_{2n} = \frac{1}{\sqrt{2}} \begin{bmatrix} \mathbf{\Omega}_n & \mathbf{F}_n \mathbf{\Omega}_n \\ \mathbf{\Omega}_n & -\mathbf{F}_n \mathbf{\Omega}_n \end{bmatrix} \mathbf{P}_{2n}.$$

   Die Fourier-Transformation eines Signals der Länge $2n$ ergibt sich also daraus, dass man es zunächst in geraden und ungerade Einträge aufteilt. Dann führt man auf beide umsortierte Hälften der Länge n die DFT rekursiv aus. Das Ergebnis des ungeraden Teils multipliziert man elementweise mit dem Vektor $(1, \omega_{2n}, \dots, \omega_{2n}^{n-1})^\mathsf{T}$. Die erste Hälfte ist die Summe der beiden Vektoren, die zweite Hälfte die Differenz.

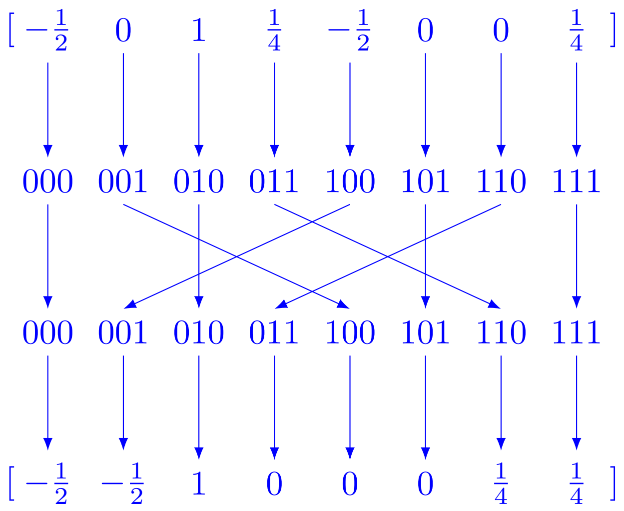
   Daraus ergibt sich ein rekursiver Algorithmus für die Fourier-Transformation, der eine bessere Laufzeit als die naive Implementierung hat.

   Lösung Ende

3. Der rekursive FFT-Algorithmus lässt sich auch interativ implementieren, wobei die erste Phase eine Umsortierung ist. Wie wird das Signal $\mathbf{s} = (s_0, \dots, s_{2^n-1})^\mathsf{T} \in \mathbb{C}^{(2^n)}$ umsortiert? Führen Sie dies explizit am Signal $\mathbf{z} = (-\frac{1}{2}, 0, 1, \frac{1}{4}, -\frac{1}{2}, 0, 0, \frac{1}{4})^\mathsf{T}$ durch.

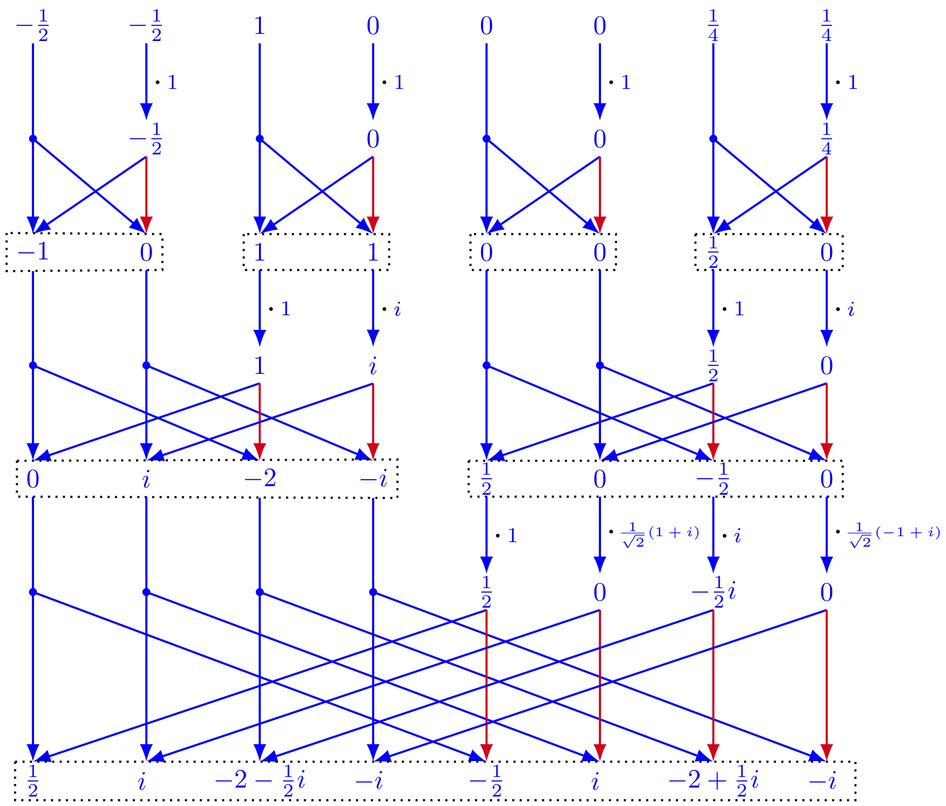
   Lösung

Die konsekutive Aufteilung des Signals in gerade und ungerade Einträge wird einmalig am Anfang der schnellen Fourier-Transformation (FFT) durchgeführt. Sie entspricht der Umkehrung der Bitreihenfolge des Indexes von LSB-0 (least significant bit) in MSB-0 (most significant bit).



Lösung Ende

4. Die nächste Phase ist die Kombinationsphase. Ergänzen Sie dazu das folgende Schema („Butterflynetzwerk"), wobei Sie mit dem umsortierten Signal aus der vorherigen Aufgabe beginnen. Geben Sie abschließend die Fourier-Transformation $\hat{\mathbf{z}}$ von $\mathbf{z}$ an, wobei Sie den Normierungsfaktor vernachlässigen dürfen.



5. Welche Laufzeit hat die schnelle Fourier-Transformation, falls man sie iterativ bzw. rekursiv implementiert?

Lösung

Die FFT hat in beiden Fällen eine Laufzeit von $\mathcal{O}(n \log n)$.

Lösung Ende

6. Wie lässt sich das Berechnungsschema so ändern, dass man die inverse Fourier-Transformation berechnet?

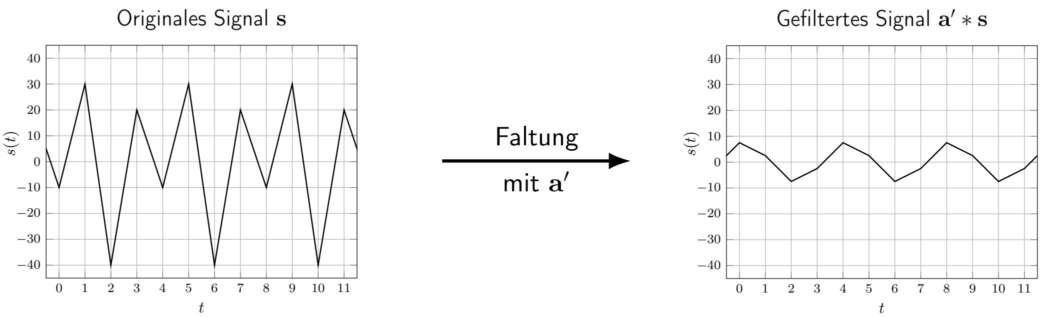
Lösung

Es bleibt die Umsortierung des Signals sowie der generelle Ablauf der Kombinationsphase identisch. Das einzige, das geändert werden muss, sind die Vorfaktoren des rechten Teils in der Kombinationsphase. Diese müssen komplex konjugiert werden.

Lösung Ende

Aufgabe 2: Faltung

Eine weitere Anwendung der schnellen Fourier-Transformation, neben der Frequenzanalyse, ist die effiziente Berechnung der diskreten, zyklischen Faltung. Gegeben ist das Signal $\mathbf{s} = (-10, 30, -40, 20)^\mathsf{T}$, das unten nochmals dargestellt ist, sowie der Faltungskern $\mathbf{a}' = \frac{1}{4}(2, 1, 0, 1)^\mathsf{T}$.



1. Die zyklische Faltung $\mathbf{a}' * \mathbf{s}$ kann man als Produkt des Signals $\mathbf{s}$ mit einer Matrix $\mathbf{A} \in \mathbb{R}^{4\times 4}$ schreiben. Geben Sie die zugehörige Faltungsmatrix $\mathbf{A}$ an.

Lösung

Die Faltungsmatrix für unseren spezifischen Faltungskern ist die folgende:

$$\mathbf{A} = \frac{1}{4}\begin{bmatrix} 2 & 1 & 0 & 1 \\ 1 & 2 & 1 & 0 \\ 0 & 1 & 2 & 1 \\ 1 & 0 & 1 & 2 \end{bmatrix}.$$

Lösung Ende

2. Wo findet sich der Faltungskern $\mathbf{a}'$ in der Matrix wieder? Wie kann man einen unbekannten Faltungskern durch die Faltung mit einem Signal ermitteln? Welche Eigenschaften hat die Faltung?

Lösung

Der Faltungskern ist die erste Spalte der Faltungsmatrix. Dieser lässt sich immer durch die Faltung $\mathbf{a}' * \mathbf{e}_1$ mit dem Signal $\mathbf{e}_1 = (1, 0, \dots, 0)$ bestimmen.

$$\begin{array}{cc} \begin{bmatrix} a_0 & a_1 & \cdots & a_{n-1} \\ a_{n-1} & a_0 & \cdots & a_{n-2} \\ \vdots & \vdots & \ddots & \vdots \\ a_1 & a_2 & \cdots & a_0 \end{bmatrix} & \begin{array}{c} \mathbf{a} \\ \\ \\ \\ \end{array} \\ \begin{array}{llll} \mathbf{a}' & & & \end{array} & \end{array}$$

Eigenschaften der Faltung:

- Linearität in beiden Argumenten (offensichtlich)
- Kommutativ: $\mathbf{a}' * \mathbf{s} = \mathbf{s} * \mathbf{a}'$
- Assoziativität und Distributivität folgen aus Linearität
- Translationsinvarianz (auch genannt Zeitinvarianz, Ortsinvarianz, Stationarität): $\mathbf{Z}^k(\mathbf{a}*\mathbf{s}) = \mathbf{a} * (\mathbf{Z}^k\mathbf{s})$, wobei $\mathbf{Z}$ die zyklische Verschiebungsmatrix ist (siehe nächste Aufgabe).

Hinweis zur Notation: Der Apostroph im Faltungskern signalisiert, dass die Indexierung der Einträge, bis auf den ersten Eintrag, umgekehrt wird. Der Vektor $\mathbf{a} = (a_0, a_1, \dots, a_{n-1})^\mathsf{T}$ wird zu $\mathbf{a}' = (a_0, a_{n-1}, \dots, a_1)^\mathsf{T}$.

Lösung Ende

3. Berechnen Sie die Faltung $\mathbf{a}' * \mathbf{s}$ im Ortsraum und zeichnen Sie das gefilterte Signal in die obige Abbildung hinein. Welchen Effekt hat die Faltung mit $\mathbf{a}'$ auf das Signal?

Lösung

Es gilt:

$$\mathbf{a}' * \mathbf{s} = \frac{1}{4}\begin{bmatrix} 2 & 1 & 0 & 1 \\ 1 & 2 & 1 & 0 \\ 0 & 1 & 2 & 1 \\ 1 & 0 & 1 & 2 \end{bmatrix} \begin{bmatrix} -10 \\ 30 \\ -40 \\ 20 \end{bmatrix} = \frac{1}{4}\begin{bmatrix} 30 \\ 10 \\ -30 \\ -10 \end{bmatrix}$$

Die Faltung mit $\mathbf{a}'$ berechnet einen gewichteten Durchschnitt jedes Signalwertes mit seinen beiden Nachbarn, wobei der Signalwert doppelt gewichtet wird. Dadurch wird das Signal geglättet.

Lösung Ende

4. Was besagt der Faltungssatz? Berechnen Sie erneut die Faltung $\mathbf{a}' * \mathbf{s}$, wenden Sie aber diesmal den Filter im Frequenzraum an.

Lösung

Der Faltungssatz besagt, dass die Faltung im Ortsraum der elementweisen Multiplikation im Frequenzraum entspricht: $\mathbf{\Omega}(\mathbf{a}'*\mathbf{s}) = \operatorname{diag}(\mathbf{\Omega}\mathbf{a}')(\mathbf{\Omega}\mathbf{s})$ (alternative Notation mit $\circ$ für elementweise Multiplikation: $\mathbf{\Omega}(\mathbf{a}' * \mathbf{s}) = (\mathbf{\Omega}\mathbf{a}') \circ (\mathbf{\Omega}\mathbf{s})$). Daraus kann die Faltung wie folgt berechnet werden: $\mathbf{a}' * \mathbf{s} = \overline{\mathbf{\Omega}}(\operatorname{diag}(\mathbf{\Omega}\mathbf{a}')(\mathbf{\Omega}\mathbf{s}))$. Im Skript findet man auch die Variante $\mathbf{a}' * \mathbf{s} = \mathbf{\Omega}(\operatorname{diag}(\overline{\mathbf{\Omega}}\mathbf{a}')(\overline{\mathbf{\Omega}}\mathbf{s}))$, wobei die DFT und die inverse DFT vertauscht werden. Folglich ist es egal, ob man die DFT oder die IDFT zum Transformieren in den Frequenzraum nutzt, man muss nur die jeweils andere als Rücktransformation verwenden.

Die DFT des Filters und des Signals sind

$$\hat{\mathbf{a}}' = \frac{1}{8}(4, 2, 0, 2)^\mathsf{T} \quad \text{und} \quad \hat{\mathbf{s}} = \frac{1}{2}(0, 30 + 10i, 0, 30 - 10i)^\mathsf{T}.$$

Nun folgt die elementweise Multiplikation: $\hat{\mathbf{a}}' \circ \hat{\mathbf{s}} = \frac{1}{16}(0, 60 + 20i, 0, 60 - 20i)$. Nun berechnet man dessen inverse Fourier-Transformation:

$$\mathbf{a}' * \mathbf{s} = \frac{1}{16}(120, 40, -120, -40) = \frac{1}{4}(30, 10, -30, -10).$$

Lösung Ende

5. Welche Laufzeit hat die zyklische Faltung, falls man sie direkt implementiert und falls man sie mithilfe des Faltungssatzes und der FFT implementiert?

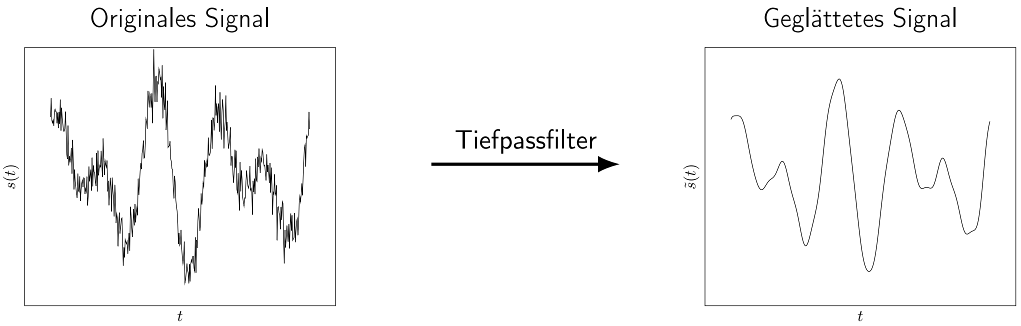
Lösung

$$\begin{tikzcd}[column sep=huge, row sep=huge]
\text{Signal } \mathbf{s}, \text{ Filter } \mathbf{a}' \arrow[r, "\text{Faltung: } \mathcal{O}(n^2)"] \arrow[d, "\text{FFT: } \mathcal{O}(n \log n)"] & \mathbf{a}' * \mathbf{s} \\
\text{Signal } \hat{\mathbf{s}}, \text{ Filter } \hat{\mathbf{a}}' \arrow[r, "\text{Elem. Mult.: } \mathcal{O}(n)"] & \hat{\mathbf{a}}' \circ \hat{\mathbf{s}} \arrow[u, "\text{IFFT: } \mathcal{O}(n \log n)"']
\end{tikzcd}$$

- Direkte Anwendung des Filters im Ortsraum: $\mathcal{O}(n^2)$
- Anwendung des Filters im Frequenzraum mit FFT: $\mathcal{O}(n \log n)$

Lösung Ende

6. Eine weitere Anwendung der schnellen Fourier-Transformation ist der aus der Hausaufgabe bekannte (ideale) Tiefpassfilter. Bei diesem werden die Anteile eines Signals, die zu einer höheren Frequenz als eine Maximalfrequenz $f_{\max}$ gehören, aus diesem entfernt.



Zwar lässt sich dieser auch durch Faltung im Ortsraum realisieren, jedoch implementiert man diesen meist über die (I)FFT.

a) Beschreiben Sie die Schritte, die notwendig sind, um die hohen Frequenzen aus dem Eingangssignal herauszufiltern.

b) Wie sieht der Filter $\hat{\mathbf{a}}'$ im Frequenzraum aus, mit dem elementweise multipliziert wird?

c) Wie sieht der zugehörige Filter $\mathbf{a}'$ im Ortsraum aus?

Lösung

a) Die Schritte lauten wie folgt:
   1) Man berechnet die Fourier-Transformation des Signals.
   2) Man setzt die Anteile zu hohen Frequenzen, unter Beachtung der Symmetrien, auf null. Das sind also die mittleren Einträge des transformierten Signals.
   3) Man berechnet die inverse Fourier-Transformation des modifizierten Frequenzspektrums.

   Dies ist äquivalent dazu, als würde man das Signal auf den Spann der ersten und letzten DFT-Basisvektoren orthogonal projizieren.

b) Der Filter sieht wie folgt im Frequenzraum aus:

$$\hat{\mathbf{a}}' = (1, 1, \dots, 1, 0, \dots, 0, 1, \dots, 1)$$

c) Der Filter im Ortsraum ergibt sich über die inverse Fourier-Transformation des Filters im Frequenzraum. Dieser ist eine Summe der ersten und letzten Spalten der (inversen) DFT-Matrix, de facto eine Superposition von langsam schwingenden Sinus- und Kosinusschwingungen.

Lösung Ende

7. Unter welchen Voraussetzungen ist es ratsam, die Faltung direkt zu implementieren, statt den Faltungssatz und die FFT zu verwenden?

Lösung

Falls der Filter oder das Signal (die Faltung ist schließlich kommutativ) dünn besetzt sind, d.h. viele Nullen beinhaltet, so ist es durchaus sinnvoll, die Faltung direkt zu implementieren.

Lösung Ende