Zur diskreten und schnellen Fouriertransformation

Vorbemerkungen

Ausgangspunkt unserer Diskussion sind periodische Funktionen: Hat $f:\mathbb{R}\to\mathbb{C}$ die Eigenschaft $\exists T\in\mathbb{R}\,\forall x\in\mathbb{R}:\ f(x)=f(x+T)$, so heißt f T-periodisch. Hat f die Periode T, so hat $x\to f(Tx)$ die Periode 1. Häufig wählt man als "generischen Fall" die Periode 2π, da die trigonometrischen Funktionen $\exp(ix)=\cos x+i\sin x$ 2π - periodisch sind. Stattdessen gehen wir im folgenden von der Periode 1 aus und nennen eine Funktion f periodisch, wenn sie diese Periode besitzt. In diesem Sinne sind für jedes $n\in\mathbb{Z}$ die Funktionen $\exp(2\pi i n x)$ periodisch. Offenbar ist eine periodische Funktion bereits durch ihre Werte auf [0,1[eindeutig bestimmt. Umgekehrt können wir eine auf [0,1[definierte Funktion periodisch auf ganz $\mathbb{R}$ fortsetzen. Also betrachten wir im folgenden nur auf [0,1[definierte Funktionen. Die Fourierkoeffizienten einer auf [0,1[integrablen Funktion f sind für $n\in\mathbb{Z}$ definiert durch $c_n=\int_0^1 f(x)\exp(-2\pi i n x)\,\mathrm{d}x$.

Umgekehrt ist f durch seine Fourierkoeffizienten bestimmt, und im Raum $L^1[0,1[$ gilt $f=\sum_{n=-\infty}^{\infty} c_n\exp(2\pi i n x)$. Interessantere Formeln erhalten wir, wenn $f\in L^2[0,1[$, da dieser Raum nicht nur wie $L^1[0,1[$ ein vollständiger normierter Vektorraum ist, sondern sogar ein hermitisches Skalarprodukt besitzt, welches die L^2 -Norm erzeugt: $<f,g>=\int_0^1 f(x)\overline{g(x)}\,\mathrm{d}x$

Die Folge $(c_n)_{n\in\mathbb{Z}}$ der Fourierkoeffizienten einer quadratintegrablen Funktion ist quadratsummierbar und wir setzen $\|c\|^2=\sum_{n=-\infty}^{\infty}|c_n|^2$. Der Raum $l^2_{\mathbb{Z}}$ der quadratsummierbaren Folgen mit komplexen Koeffizienten ist selbst ein Vektorraum mit einem hermitischen Skalarprodukt, welches durch $<c,d>=\sum_{n=-\infty}^{\infty} c_d\overline{d_n}$ gegeben ist. Sind $f,g\in L^2[0,1[$ und $(c_n),(d_n)$ die zugehörigen Folgen der Fourierkoeffizienten, so läßt sich sogar zeigen, daß $<f,g>=<c,d>$. Insbesondere gilt $\|f\|=\|c\|$.

Will man konkret mit einer Funktion f auf [0,1[rechnen, so kann man sie "diskretisieren", d.h. nur auf endlich vielen Punkten des Intervalls auswerten. Wählt man die N Punkte $x_0=0, x_1=\frac{1}{N},\ldots,x_{N-1}=\frac{N-1}{N}$ und die zugehörigen Funktionswerte $a_0,\ldots,a_{N-1}$, so werden die Fourierkoeffizienten $c_n=\int_0^1 f_n(x)\exp(-2\pi i nx)\,\mathrm{d}x$ approximiert durch die Riemann-Summe $b_n=\frac{1}{N}\sum_{k=0}^{N-1} a_i\exp\left(-2\pi i\frac{kn}{N}\right)$. Offenbar macht es keinen Sinn, Koeffizienten b_n mit $n\geqslant N$ zu betrachten, da dann die Schwingungen der $\exp(-2\pi i nx)$ feiner werden als die gewählte "Auflösung" durch die N Intervallpunkte an Funktionswerten für f hergibt. Man rechnet auch leicht nach, daß für diese "diskretisierten" Fourierkoeffizienten gilt $b_{n+N}=b_n$. Es reicht daher, die Koeffizienten $b_0,\ldots,b_{N-1}$ zu betrachten. Umgekehrt erhält man die Funktionswerte $f(x_n)=a_n$ zurück durch die Transformation $a_n=\sum_{k=0}^{N-1} b_k\exp\left(2\pi i\frac{kn}{N}\right)$, dies war in Blatt 4, Aufg. 1 zu zeigen.

Schnelle Fouriertransformation

Wie man oben sieht, sind bei der Berechnung der diskreten Fouriertransformation bzw. deren Inverser n^2 Multiplikationen nötig. Es ist unbedingt notwendig, diesen Aufwand zu reduzieren.

Zunächst setzen wir für $n \in \mathbb{N}_0$ $\omega_n = \exp\left(-2\pi i \frac{1}{2^n}\right)$. Man sagt, ω_n sei eine "primitive 2^n - te komplexe Einheitswurzel". Offenbar gilt $\omega_n^2 = \omega_{n-1}$ und $\omega_n^{2^n} = 1$.

Für $0 \leqslant j < 2^n$ möchten wir zu einem gegebenen Vektor $a = \begin{pmatrix} a_0 \\ \vdots \\ a_{2^n-1} \end{pmatrix} \in \mathbb{C}^{2^n}$ die Werte

$d_j = \sum_{k=0}^{2^n-1} a_k \exp\left(2\pi i \frac{kj}{2^n}\right)$ und damit den Vektor $d = \begin{pmatrix} d_0 \\ \vdots \\ d_{2^n-1} \end{pmatrix} \in \mathbb{C}^{2^n}$ berechnen. Mit obiger Einheitswurzel-Notation können wir einfacher schreiben: $d_j = \sum_{k=0}^{2^n-1} a_k \omega_n^{jk}$. Wir haben in Blatt 4 Aufg. 1 die lineare Transformation $\mathcal{F}_n : \mathbb{C} \to \mathbb{C}$ $\mathcal{F}_n(a) = d$ näher untersucht und z.B. ihre Matrixdarstellung und ihre Inverse berechnet.

Wir können für $n > 0$ die Summe $d_j = \sum_{k=0}^{2^n-1} a_k \omega_n^{jk}$ aufspalten und erhalten unter Ausnutzung von $\omega_n^{2k} = \left(\omega_n^2\right)^k = \omega_{n-1}^k$ $\sum_{k=0}^{2^n-1} a_k \omega_n^{jk} = \sum_{k=0}^{2^{n-1}-1} a_{2k} \omega_{n-1}^{jk} + \omega_n^j \sum_{k=0}^{2^{n-1}-1} a_{2k+1} \omega_{n-1}^{jk}$. Wenn man die Summen auf der rechten Seite als Fouriertransformationen von nur noch halb so langen Vektoren betrachtet, muß man auf der rechten Seite den Exponenten j durch $j_{n-1} = j \% 2^n$ ersetzen. Für $2^{n-1} \leqslant j < 2^n$ ist nämlich $j = 2^{n-1} + j_{n-1}$ und daher $\omega_{n-1}^{jk} = \left(\omega_{n-1}^{2^{n-1}+j_{n-1}}\right)^k = \left(\omega_{n-1}^{2^{n-1}} \cdot \omega_{n-1}^{j_{n-1}}\right)^k = \omega_{n-1}^{j_{n-1}k}$

Mit den Abbildungen $g, u : \mathbb{C}^{2^n} \to \mathbb{C}^{2^{n-1}}$, $g(a) = \begin{pmatrix} a_0 \\ a_2 \\ a_4 \\ \vdots \\ a_{2^n-2} \end{pmatrix}$, $u(a) = \begin{pmatrix} a_1 \\ a_3 \\ a_5 \\ \vdots \\ a_{2^n-1} \end{pmatrix}$ folgt jetzt:

$$\mathcal{F}_n(a)_{j_n} = \mathcal{F}_{n-1}(g(a))_{j_{n-1}} + \omega_n^j \mathcal{F}_{n-1}(u(a))_{j_{n-1}} .$$

Diese Formel kann offenbar als Basis einer schnellen rekursiven Berechnung von $\mathcal{F}_n$ dienen.

In einem Computerprogramm würde dies bedeuten, daß die Funktion sich selbst mehrfach aufruft, was einen nicht unbeträchtlichen Overhead schafft. Um dies zu vermeiden, könnte man so vorgehen:

Zunächst idenfiziere man Zahlen mit den Bistrings ihrer Dualdarstellung, d.h. $j_{k-1} j_{k-2} \ldots j_0 = \sum_0^{k-1} j_k 2^k$. Damit ist ein zu transformierender Vektor $a \in \mathbb{C}^{2^n}$ gegeben durch seine Komponenten $a_{j_{n-1}\cdots j_0}$.

Wir definieren rekursiv $a^{i_s\cdots i_0}_{j_r\cdots j_0} = a^{i_{s-1}\cdots i_0}_{0\,j_r\cdots j_0} + \omega^{i_s\cdots i_0}_{s+1}\, a^{i_{s-1}\cdots i_0}_{1\,j_r\cdots j_0}$ und erhalten $a^{i_{n-1}\cdots i_0}$ als Komponenten des transformierten Vektors.

Bei der inversen Transformation sind lediglich die benutzen Einheitswurzeln ω_k durch ihre Konjugierten zu ersetzen und das Ergebnis mit dem Normierungsfaktor $\frac{1}{2^n}$ zu multiplizieren.