JPEG-2000: de wondere wereld van Wavelets

Jan Westerdiep Ogier van Garderen

23 juni 2013

Inhoudsopgave

Hoofdstuk 1

Intro

1.1 Signaaluitbreiding

Beide algoritmes kunnen enkel omgaan met signalen die een tweemacht lang zijn. Om te zorgen dat een willekeurig signaal ook getransformeerd kan worden, moet het dus uitgebreid worden voorbij zijn definitiegebied. De meeste bronnen onderscheiden de volgende manieren om het signaal uit te breiden. Laat $x_1, x_2, \ldots x_n$ het signaal.

Zero-padding $x' = 0, \ldots, 0|x_1, x_2, \ldots, x_n|0, \ldots, 0$. Het gevolg van deze uitbreiding is dat het signaal niet langer continu hoeft te zijn op de randen;

Constant padding $x' = x_1, \ldots, x_1|x_1, x_2, \ldots, x_n|x_n, \ldots, x_n$. Het gevolg is nu dat het signaal geen continue afgeleide meer hoeft te hebben;

Symmetric padding $x' = x_n, \ldots, x_1|x_1, x_2, \ldots, x_n|x_n, \ldots, x_1$. Het gevolg is nu dat het signaal geen continue afgeleide meer hoeft te hebben;

Periodic padding $x' = x_1, \ldots, x_n|x_1, x_2, \ldots, x_n|x_1, \ldots, x_n$. Het nieuwe signaal hoeft wederom niet continu te worden.

Wij hebben ervoor gekozen om signalen uit te breiden door middel van zero-padding.

Hoofdstuk 2

Fourier

Studie van signalenperiodieke functies / een manier om een functie te ontbinden in terugkerende patronen / karakteristiek van functie in termen van bepaalde frequenties / overtuigendere manier om 'nette' functies te beschrijven / multiresolutie in de tijd

TODO: Periodieke functies zijn ook wel te beschrijven door slechts naar één fase te kijken.

De basisfuncties waarmee we een functie karakteriseren bij Fourieranalyse worden gegeven door de complexe e-machten.

Definitie (FourierBasis)**.** *Zij gegeven een interval* $[a, b]$*, en bekijk de functieruimte* $L_2([a, b])$*. Laat de verzameling* $F_{[a,b]}$ *gedefinieerd zijn door:*

$$F_{[a,b]} := \left\{ \phi_k(x) = \tfrac{1}{\sqrt{b-a}} e^{2\pi i \cdot k \frac{x-a}{b-a}} \;\middle|\; k \in \mathbb{N}_0 \right\}$$

Dan noemen we $F_{[a,b]}$ *de Fourier'basis' van* $L_2([a, b])$

Merk op dat hier het woord *basis* nog tussen aanhalingstekens staat want we zullen eerst moeten bewijzen dat deze verzameling ook echte een (complexe) orthonormale basis vormt voor $L_2([a, b])$.

Ten eerste zullen we zien dat alle elementen van $F_{[a,b]}$ orthonormaal zijn t.o.v. het complexe inproduct van functies en zijn geïnduceerde norm:

$$\langle \phi_k, \phi_j \rangle = \frac{1}{(b-a)} \int_{[a,b]} e^{\frac{2\pi i}{b-a} xk} e^{-\frac{2\pi i}{b-a} xj} \mathrm{d}x = \frac{1}{2\pi} \int_{[0,2\pi]} e^{iy(k-j)} \mathrm{d}y = \begin{cases} 0 & \text{als } k \neq j, \\ 1 & \text{als } k = j. \end{cases}$$

Hieruit volgt dat de elementen van $F_{[a,b]}$ een orthonormale basis vormen voor de ruimte $\tilde{F} = \mathrm{span}(F_{[a,b]})$. We willen nu bewijzen dat de reële projectie van deze ruimte dicht ligt in de reële functieruimte $L_2([a, b])$. We zullen hier eerst een aantal handige definities voor introduceren.

Definitie (Fouriergetransformeerde)**.** *Zij gegeven een functie $f \in L_2([a,b])$, schrijf dan een vector $\hat{f} \in \mathbb{C}^\infty$ met entries gedefinieerd volgens (hier aangegeven met blokhaken):*

$$\hat{f}[n] = \frac{1}{\sqrt{b-a}} \cdot \langle f, \phi_n \rangle = \frac{1}{b-a} \int_{[a,b]} f(x) \cdot e^{-2\pi i \cdot k \frac{x-a}{b-a}} \, dx.$$

We noemen $\hat{f}$ de Fouriergetransformeerde van f.

TODO: Bewijzen dat alle reële functies in de ruimte opgespannen door de Fourier-basis dicht liggen in L_2.

TODO: Bewijzen dat de continue reconstructie perfect is wanneer $f \in C^1$

2.1 De *Discrete Fourier Transform*

Zoals we gezien hebben in het vorige hoofdstuk kan de Fouriertransformatie gebruikt worden om continue signalen te karakteriseren voor verschillende frequenties. Een groot gebied binnen de signaalanalyse is echter van discrete aard aangezien hier toch veelal digitale instrumenten voor worden gebruikt. De toepassingen waar wij op zoek zijn liggen ook in dit digitale domein (JPEG is immers een beeldcompressie-algoritme) en dus zal ons verslag zich verder afspelen in deze discrete setting.

Om discrete signalen te analyseren lijkt het in eerste instantie voor de hand te liggen om het discrete signaal als stapfunctie te bekijken, deze is immers ook kwadratisch integreerbaar. Dit leidt echter tot ongewenste resultaten, zoals is te zien aan de reconstructie van een blokgolf door middel van Fouriertransformatie. Erger nog, elke eindige som van continue functies is weer continu dus voor een perfecte reconstructie van een discreet signaal is met deze methode altijd een oneindige rij coëfficiënten nodig. Bovendien is het moeilijk om uit deze coëfficiënten relevante informatie te destilleren over de aard van het signaal.

In plaats van de discrete signalen in te bedden in L_2 zullen we een discreet analogon gebruiken om de Fouriertransformatie van praktisch nut te laten zijn in de analyse van discrete signalen. Hiervoor zullen we de Fourier-basis moeten discretiseren en ons moeten richten op de ruimte $\mathbb{R}^n$. We doen dit op zo'n manier dat deze discretisatie in de limiet naar het continue geval overgaat. Zo zullen we uiteindelijk tot de *Discrete Fourier Transform* komen.

Definitie (Discrete Fourierbasis)**.** *Gegeven de ruimte $\mathbb{R}^n$, definieer dan de verzameling*

$$S_n := \left\{ s_k[j] = e^{2\pi i k j/n} \,\middle|\, k, j \in \{0, \ldots, n-1\} \right\}$$

Als zijnde de discrete Fourierbasis met basisvectoren s_k.

Met deze basis in de hand kunnen we vervolgens de discrete Fouriergetransformeerde (DFT) van een signaal schrijven als de vector van (complexe) inproducten van dit signaal met de

DFT-basisfuncties. Laat voor een signaal x van lengte n de DFT (aangeduid met X) gegeven zijn door:

$$X[k] = \tfrac{1}{n}\langle x, s_k\rangle \quad k \in \{0, \ldots, n-1\}$$

Ook definiëren we een operatie op zulke vectoren X en beweren dat deze een inverse DFT – ook wel iDFT – voorstelt (dit bewijs volgt weldra in een volgende sectie). De iDFT wordt gegeven door

$$x[j] = \langle X, s_j^{-1}\rangle \quad j \in \{0, \ldots, n-1\}$$

We zullen nu enkele bewijzen geven om de term 'Discrete Fourierbasis' te ondersteunen.

Stelling (Fourierbasis)**.** *De vectoren s_k zoals gedefinieerd voor de verzameling S_k staan onderling loodrecht onder het complexe inproduct en S_k is dus een complexe basis voor $\mathbb{C}^n$.*

Bewijs. Bekijk het complexe inproduct tussen twee willekeurig gekozen elementen van deze verzameling:

$$\langle s_k, s_j\rangle = \sum_{m\leq n} s_k[m]\cdot\overline{s_j[m]} = \sum_{m\leq n} e^{2\pi i\cdot m(k-j)/n} = \begin{cases} 0 & \text{als } k \neq j \\ n & \text{als } k = j \end{cases}$$

Dus staan de vectoren van S_n onderling loodrecht, zodat deze een complexe ruimte opspannen van dimensie n. □

Om de claim te ondersteunen dat de DFT-methode een echte discretisatie is van de continue Fouriertransformatie willen we bewijzen dat dit algoritme voor een steeds fijnere selectie van waardes van een functie in de limiet hetzelfde resultaat geeft als de continue Fouriertransformatie.

Zoals gebruikelijk bij het overschakelen van een discrete naar een continue setting, kunnen we dit doen door de definitie van de Riemann integraal toe te passen op de sommatie die voor handen ligt.

Stelling (Limiet van discrete Fourier-transformatie)**.** *Gegeven een interval $[a, b]$ en een functie $f \in L_2([a, b])$. Voor een discretisatie van f in n gelijke intervallen zodat de discrete f precies één waarde uit elk interval inneemt geldt dat de discrete Fouriergetransformeerde limiteert naar de algemene Fouriergetransformeerde wanneer $n \to \infty$.*

Bewijs. Gegeven een interval $[a, b]$ kunnen we een partitie P maken in n gelijke delen, ofwel laat $P = \{a = t_0, t_1, .., t_n = b\}$ met $t_j = \frac{na+j(b-a)}{n}$. We discretiseren onze functie f door uit elk interval $[t_{j-1}, t_j]$ van de partitie de waarde in $x_j = t_j - c_j(\frac{b-a}{n})$ te nemen met $c_j \in [0, 1]$, ofwel $f[j] = f(\frac{b-a}{n}(j - c_j) + a)$. De DFT van de discrete functie $f[\cdot]$ wordt dan gegeven door het inproduct te nemen met de DFT-basisfuncties:

$$F[k] = \frac{1}{n}\sum_{j\leq n} f[j]\cdot\overline{s_k}[j].$$

We schrijven dit om in termen van onze partitie, met de identificatie $n \cdot \frac{x_j - a}{b-a} + c_j = j$ waarmee we de continue variable $x_j \in [a, b]$ omschrijven naar zijn discrete tegenhanger en krijgen zo:

$$\begin{aligned} F[k] = & \quad \tfrac{1}{n} \textstyle\sum_{j \leq n} f[j] \cdot \overline{s_k}[n \cdot \frac{x_j - a}{b-a} + c_j] \\ = & \quad \tfrac{1}{n} \textstyle\sum_{j \leq n} f(x_j) \cdot e^{-2\pi i \cdot k \frac{x-a}{b-a}} \cdot e^{-2\pi i \cdot k c_j / n} \\ = & \quad \tfrac{1}{\sqrt{b-a}} \textstyle\sum_{j \leq n} f(x_j) \cdot \overline{\phi}_k(x_j) \cdot \frac{b-a}{n} \cdot e^{-2\pi i \cdot k c_j / n} \end{aligned}$$

We merken op dat c_j begrensd dus we weten dat de limiet van $e^{-2\pi i \cdot k c_j / n}$ voor $n \to \infty$ gelijk is aan 1 voor elke j. Dan is de limiet voor $F[k]$ gelijk aan de limiet van de overige factoren in de sommatie. We merken op dat de term $\frac{b-a}{n}$ precies de grootte is van ons interval en dat we het geheel omgeschreven hebben in termen van onze continue functies f en ϕ_k. Omdat het product $f \cdot \phi_k$ integreerbaar is moet voor elke partitie P met waarden in punten x_j uit elk interval deze sommatie convergeren naar de integraal

$$\frac{1}{\sqrt{b-a}} \int_{[a,b]} f(x) \overline{\phi}_k(x) dx$$

wanneer we de maaswijdte ($\frac{b-a}{n}$) naar 0 laten gaan. Dit is duidelijk het geval wanneer we de limiet $n \to \infty$ nemen. Daarmee is de DFT een goede discretisatie van de Fouriergetransformeerde. □

Voor de werking van het DFT algoritme als signaalcompressie-algoritme is het van belang dat er een inverse algoritme bestaat dat het getransformeerde inputsignaal weer terugtransformeert zonder verlies van informatie. We zullen nu bewijzen dat dit mogelijk is door de DFT en iDFT te gebruiken.

Stelling (Inverteerbaarheid van de DFT)**.** *De samenstelling van de inverse Discrete Fouriertransformatie met de Discrete Fouriertransformatie is gelijk aan de identiteit.*

Bewijs. We bekijken daarvoor een signaal $x \in \mathbb{R}^n$ dat met de DFT methode getransformeerd wordt in een vector $X \in \mathbb{C}^n$. Hierbij hoort de discrete Fourier-basis S_n met basisvectoren s_k. De discrete Fourier transformatie en zijn inverse zijn gedefiniëerd door:

$$\begin{aligned} X[k] = & \quad \tfrac{1}{n} \langle x, s_k \rangle \qquad \text{(DFT)} \\ \hat{x}[j] = & \quad \langle X, s_j^{-1} \rangle \qquad \text{(iDFT)} \end{aligned}$$

Om te bewijzen dat de iDFT de DFT inverteert voldoet het om de iDFT uit te schrijven in termen van x door X in te vullen volgens de formule van de DFT. Uitschrijven hiervan geeft:

$$\begin{aligned} \hat{x}[j] = & \quad \langle X, s_j^{-1} \rangle \\ = & \quad \textstyle\sum_{i=1}^n X[i] \cdot \overline{s_j^{-1}}[i] \\ = & \quad \textstyle\sum_{i=1}^n \frac{1}{n} \langle x, s_i \rangle \overline{s_j^{-1}}[i] \\ = & \quad \langle x, \frac{1}{n} \textstyle\sum_{i=1}^n s_i s_j^{-1}[i] \rangle \end{aligned}$$

We schrijven nu de sommatie in de tweede term van het inproduct (die een vector van lengte n representeert) uit in zijn componenten. Merk op dat de discrete Fourier-basisvectoren zo

gedefinieerd zijn dat $s_k[m] = s_m[k]$. We verwisselen de indices in blokhaken met die in het subscript van s en verkrijgen daarmee

$$\frac{1}{n}\sum_{i=1}^{n} s_i[m]s_j^{-1}[i] = \frac{1}{n}\sum_{i=1}^{n} s_m[i]\overline{s_j}[i] = \tfrac{1}{n}\langle s_m, s_j\rangle = \begin{cases} 0 & \text{als } m \neq j \\ 1 & \text{als } m = j \end{cases} .$$

Hier gebruiken we dus de eerdere vergelijking voor het inproduct van twee basisfuncties. We gebruiken deze eigenschap van de sommatie om ons eerder inproduct uit te schrijven. We hebben nu een gelijkheid

$$\hat{x}_j = \langle x, e_j\rangle = x_j$$

Door de iDFT toe te passen op het resultaat van de DFT zullen we altijd het origineel terug krijgen, dus de iDFT samengesteld met de DFT geeft een identiteit. □

2.2 De Fast Fourier Transform

De snelheid van het DFT algoritme valt in de praktijk nogal tegen, het nemen van n inproducten over vectoren van lengte n heeft namelijk een tijdscomplexiteit van $\mathcal{O}(n^2)$. Dit staat de directe implementatie van de DFT voor praktische toepassingen in de weg. Daarom is er een alternatief algoritme, de *Fast Fourier Transform*.

Algoritme (Fast Fourier Transform)**.** *Gegeven zij een inputsignaal x van lengte $n = 2^m$, dan geeft het algoritme FFT een lijst terug van waardes X van lengte $n = 2^m$ als volgt:*

Als $m = 0$ dan geeft de FFT de lijst (van één element) direct terug:

$$X = x.$$

Wanneer $m \neq 0$ splitsen we de lijst x op in lijsten ϵ, o van zijn even en oneven indices:

$$\begin{aligned} \epsilon[k] = &\quad x[2k] &&\text{voor } k < n/2 \\ o[k] = &\quad x[2k+1] &&\text{voor } k < n/2 \end{aligned}$$

Vervolgens voeren we hierop het FFT algoritme uit om de volgende lijsten te verkrijgen:

$$\begin{aligned} E = &\quad FFT(\epsilon) \\ O = &\quad FFT(o) \end{aligned}$$

Hiermee wordt de output van het algoritme geconstrueerd als volgt:

$$X[k] = \begin{cases} E[k] & + & e^{-2\pi ik/n} \cdot O[k] & k < n/2 \\ E[k-n/2] & - & e^{-2\pi i(k-n/2)/n} \cdot O[k-n/2] & k \geq n/2 \end{cases}$$

Dit is dus een recursief gedefinieerd algoritme dat een signaal meermaals halveert en in zichzelf terugvoert. Het is gegarandeerd dat dit algoritme afloopt vanwege de conditie op $n = 0$ samen met de halvering van de input bij elke stap. Een belangrijke voorwaarde voor de relevantie van de FFT is nu dat het algoritme hetzelfde resultaat geeft als het DFT algoritme en dit zullen we nu dan ook bewijzen.

Stelling. *Het uitvoeren van het Fast Fourier Transform algoritme op een dataset van lengte $n = 2^m$ geeft dezelfde getransformeerde als de discrete Fouriertranformatie.*

Bewijs. We gebruiken hier een inductief bewijs met inductie naar n. Onze aanname is dat het FFT-algoritme voor x van lengte $n = 2^m$ gelijk is aan de DFT van x, ofwel

$$X[k] = \sum_{k=1}^{N} x[j] \cdot e^{-2\pi i \cdot jk/n}$$

Dit geldt duidelijkerwijs wanneer $m = 0$, onze basistap. Hiervoor geldt namelijk:

$$X[k] = x[k] = x[1] = \sum_{k=1}^{2^0} x[j] \cdot e^{-2\pi i \cdot 1/2^0}$$

Vervolgens passen we inductie toe naar m door onze aanname voor $m-1$ te gebruiken, we vullen hiermee $E[k]$ en $O[k]$ in de vergelijking voor $X[k]$ in, deze hebben immers lengte $n = 2^{m-1}$.

$$X[k] = \begin{cases} \sum_{j=1}^{n/2} \epsilon[j] \cdot e^{-2\pi i \cdot kj \cdot 2/n} & + \quad e^{-2\pi i \cdot k/n} \sum_{j=1}^{n/2} o[j] \cdot e^{-2\pi i \cdot kj \cdot 2/n} & k < n/2 \\ \sum_{j=1}^{n/2} \epsilon[j] \cdot e^{-2\pi i \cdot (k-n/2)j \cdot 2/n} & - \quad e^{-2\pi i \cdot (k-n/2)/n} \sum_{j=1}^{n/2} o[j] \cdot e^{-2\pi i \cdot (k-n/2)j \cdot 2/n} & k \geq n/2 \end{cases}$$

We merken op dat we de e-machten in het tweede geval kunnen vereenvoudigen volgens

$$e^{-2\pi i \cdot (k-n/2)j \cdot 2/n} = -e^{-2\pi i \cdot kj \cdot 2/n} \quad , \quad e^{-2\pi i \cdot (k-n/2)/n} = -e^{-2\pi i \cdot k/n},$$

waardoor het gevalsonderscheid wegvalt, aangezien beide vergelijkingen nu identiek zijn. We verkrijgen $X[k]$ als sommatie over de lijsten e en o, we vullen de relatie voor e, o met x in, en nemen de factor voor de oneven indices mee in de sommatie.

$$X[k] = \sum_{j=1}^{n/2} x[2j] \cdot e^{-2\pi i \cdot k(2j)/n} + \sum_{j=1}^{n/2} x[2j+1] \cdot e^{-2\pi i \cdot k(2j+1)/n} = \sum_{j=1}^{n} x[j] \cdot e^{-2\pi i \cdot kj/n}$$

Dit bewijst dat de FFT hetzelfde resultaat levert als het DFT algoritme, het bewijs voor de gelijkheid van de iDFT en de inverse FFT is hetzelfde wanneer men de substitutie $-2\pi i \to 2\pi i$ uitvoert. □

Opmerking. We hebben hier telkens aangenomen, en zullen deze aanname ook doorzetten, dat de lengte van het ingangssignaal een macht van 2 is. Dit is een belangrijke eigenschap waar de variant van het FFT-algoritme dat hier gebruikt wordt door werkt. Deze versie van FFT wordt de Radix-2 Decimation In Time van het Cooley-Tukey FFT algoritme genoemd. Algemenere vormen van het algoritme worden ook toegepast in geoptimaliseerde algoritmes maar om de implementatie te versimpelen is voor Radix-2 gekozen. Eventuele verschillen in afmetingen tussen een signaal en een 2-macht zijn opgelost met signaalextensie, zoals eerder beschreven.

Complexiteit van de Fast Fourier Transform

Hoewel het vanuit een puur wiskundig oogpunt wellicht minder relevant is, is de complexiteit van een algoritme als een factor van belangrijk praktisch nut. De complexiteit vertaalt namelijk direct naar de looptijd van het algoritme. We hebben hier de volgende stelling uit de complexiteitstheorie nodig. [1]

Stelling (Akra-Bazzi)**.** *Zij $T(n)$ een recurrente betrekking van de vorm*

$$T(n) = \begin{cases} c & \text{als } n \leq d \\ aT(n/b) + f(n) & \text{anders,} \end{cases}$$

waarbij $a, b, d \in \mathbb{N}$, $c \in \mathbb{R}$ en f een functie $f : \mathbb{N} \to \mathbb{R}$ die voldoet aan

$$\exists k \in \mathbb{N} \,:\, f(n) \in \theta(n^{\log a / \log b} \log^k n).$$

Dan wordt de orde van $T(n)$ gegeven door:

$$T(n) \in \theta(n^{\log a / \log b} \log^{k+1} n).$$

We beschouwen hier $T(n)$ als het aantal stappen dat een machine nodig heeft om het algoritme uit te voeren. Deze stelling is voldoende om een uitspraak te kunnen doen over de complexiteit

Stelling (Complexiteit van de FFT)**.** *Het Fast Fourier Transform algoritme heeft een tijdscomplexiteit $\mathcal{O}(n \log n)$ voor input van lengte $n = 2^m$*

Bewijs. We schrijven het FFT algoritme in pseudocode.

```
function FFT(x)
    n ← lengte(x)                          ▷ Assumptie: n = 2^m voor een m
    if n == 1 then
        X ← x
    else
        E ← FFT(x[0 :: 2])                 ▷ FFT op even indices
        O ← FFT(x[1 :: 2])                 ▷ FFT oneven indices
        for i = 0 to n − 1 do
            if i < n/2 then
                X[i] ← E[i] + e^{−2iπk/n} · O[i]
            else
                X[i] ← E[i] − e^{−2iπk/n} · O[i]
            end if
        end for
    end if
    return X
end function
```

[1] Dit is een speciaal geval van de stelling, voor de volledige stelleing en het bewijs zie [1].

Dit algoritme is recursief, dus kunnen we de complexiteit schrijven door middel van een recurrente betrekking. Laat hiervoor $T(n)$ het aantal berekeningen zijn dat het algoritme kost bij een invoersignaal van lengte n. We maken een gevalsonderscheid: als de lijst lengte 1 heeft geven we deze direct terug (1 berekening). Bij een lijst van lengte > 1 splitsen we de lijst op in de even en oneven entries en voeren we op beiden weer het FFT algoritme uit, vervolgens voeren we nog n maal een vast aantal (c) berekeningen uit om tot het eindresultaat te komen. In formulevorm geeft dit de *recurrente betrekking*

$$T(n) = \begin{cases} 1 & \text{als } n = 1 \\ 2 \cdot T(n/2) + c \cdot n & \text{anders.} \end{cases}$$

We zullen nu bovenstaande (vereenvoudigde) stelling van Akra-Bazzi gebruiken. We rekenen hierbij niet de $\mathcal{O}$ van de FFT uit maar de strictere θ die gedefinieerd is volgens:

$$f \in \theta(g) \Leftrightarrow \exists c \in \mathbb{R}_+ : \lim_{n\to\infty} \frac{f(n)}{g(n)} = c$$

Zonder verder al te veel in te gaan op de implicaties die θ heeft op het gedrag van de FFT gebruiken we dan dat in ieder geval geldt dat $f \in \theta(g)$ impliceert dat $f \in \mathcal{O}(g)$.

De recurrente betrekking voor de complexiteit van de FFT is inderdaad van dezelfde vorm als die van $T(n)$ in bovenstaande stelling. Laat hiervoor namelijk $a = b = 2$, $c = d = 1$ en f de functie die $n \mapsto n$ dan voldoet f aan:

$$f(n) \in \theta(n^{\log 2/\log 2} \log^0 n) = \theta(n).$$

Dit betekent dat $T(n) \in \theta(n \log n)$ en dus zeker $T(n) \in \mathcal{O}(n \log n)$. Hiermee hebben we bewezen dat de FFT en daarmee de iFFT binnen tijdscomplexiteit $\mathcal{O}(n \log n)$ lopen. □

2.3 Discrete Fourier Transform in meer dimensies

Een eigenschap van het DFT-algoritme is dat het op een natuurlijke manier uit te breiden is naar hogere dimensies. Per extensie daarvan is er een manier om het FFT-algoritme mee te laten schalen, die we ook zullen behandelen. Het idee hierbij is om het Tensorproduct te nemen van meerdere bases.

Definitie (Multidimensionale Discrete Fourierbasis)**.** *Gegeven zij een signaalruimte van de vorm $\mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \times \ldots \times \mathbb{R}^{n_m}$ welke we opvatten als een m-dimensionaal discreet signaal waarbij de i-de orthogonale richting een lengte n_i heeft. Dan definiëren we de multidimensionale discrete Fourier'basis' behorende bij deze signaalruimte als*

$$S_{\mathbf{n}} = S_{n_1} \otimes S_{n_2} \otimes \ldots \otimes S_{n_m} = \left\{ s_{\mathbf{k}}[\mathbf{j}] = s_{k_1}[j_1] \cdot s_{k_2}[j_2] \cdot \ldots \cdot s_{k_m}[j_m] \;\middle|\; s_{k_i} \in S_{n_i}, j_i \in \{1, \ldots, n_i\} \right\}$$

Met basisvectoren $s_{\mathbf{k}}$, waar $\mathbf{k}$ een indexvector uit $\{0, \ldots, n_1\} \times \ldots \times \{0 \ldots, n_m\}$ is.

Wanneer we nu een m-dimensionaal signaal bekijken van lengte n_1 bij n_2 etc., dan kunnen we hierop een multidimensionale Fouriertransformatie op definiëren door het inproduct te

generaliseren naar onze m-dimensionale signaalruimte.

$$X[\mathbf{k}] = \tfrac{1}{\mathbf{n}} \sum_{\mathbf{j}=\mathbf{1}}^{\mathbf{n}} x[\mathbf{j}]\overline{s_{\mathbf{k}}}[\mathbf{j}] = \left(\prod_{i \leq m} \tfrac{1}{n_i} \right) \cdot \sum_{j_1=1}^{n_1} \cdots \sum_{j_m=1}^{n_m} x[j_1, \ldots, j_m] \cdot e^{-2\pi i \cdot k_m \cdot j_m / n_m} \cdot \ldots \cdot e^{-2\pi i \cdot k_1 \cdot j_1 / n_1}$$

We zullen echter een andere definitie gebruiken die equivalent is aan deze method, maar recursief gedefinieerd. We noteren deze transformatie als DFT_m, een functie die een m-dimensionaal ingangssignaal transformeert.

Definitie (Multidimensionale Discrete Fouriertransformatie)**.** *Definieer DFT_0 als de identiteit op $\mathbb{R}^{n_1}$. We definiëren dan een DFT van orde m (notatie DFT_m) als een transformatie $\mathbb{R}^{n_1} \times \ldots \times \mathbb{R}^{n_m} \to \mathbb{C}^{n_1} \times \ldots \times \mathbb{C}^{n_m}$ aan de hand van het beeld X van een m-dimensionaal ingangssignaal x:*

$$X[k_1, \ldots, k_m] = \tfrac{1}{n_m} \sum_{j_m=1}^{n_m} DFT_{m-1}(x[j_m])[k_1, \ldots, k_{m-1}] \cdot s_{k_m}[j_m]$$

Hier staat de notatie $x[j_m]$ voor een $(m-1)$-dimensionaal signaal dat we verkrijgen door de coördinaat in de m-de richting vast te zetten.

Dit is duidelijk dezelfde definitie als eerder gegeven in termen van herhaalde sommatie. Voor $m = 1$ staat er immers de gewone DFT en voor hogere machten schaalt dit met een factor $\frac{1}{n}$ en een extra sommatie. Het voordeel van deze definitie is dat het makkelijk om te schrijven is naar een algoritme dat herhaald DFT toepast.

Algoritme (Multidimensionaal DFT-algoritme)**.** *Gegeven is een m-dimensionale input x en een huidig level t, we schrijven het algoritme met output X_t als:*

$$\begin{aligned} X_t[k_1, \ldots, k_{t-1}, k_{t+1}, \ldots, k_m] &= DFT(X_{t-1}[k_1, \ldots, k_{t-1}, k_{t+1}, \ldots, k_m]) \\ &\text{voor } k_1 \leq n_1, \ldots, k_{t-1} \leq n_{t-1}, k_{t+1} \leq n_{t+1}, \ldots k_m \leq n_m \end{aligned}$$

Waarbij we de randvoorwaarde $X_0 = x$ gebruiken. We beweren dat dit algoritme de multidimensionale Fouriertransformatie van een m-dimensionaal signaal geeft.

Bewijs. Om te bewijzen dat dit de MDFT geeft volstaat het om aan te tonen dat

$$X_t[k_1, \ldots, k_m] = \sum_{j_t=1}^{n_t} \mathrm{DFT}_{t-1}(x[j_t, k_{t+1}, \ldots, k_m])[k_1, \ldots, k_{t-1}] \cdot s_{k_t}[j_t]$$

Wanneer dit geldt voor elke t dan geldt dit namelijk in het bijzonder voor het geval $t = m$, zodat X_m gegeven wordt voor de formule voor de MDFT.

Laat $t = 1$ dan geldt:

$$\begin{aligned} X_1[k_1, \ldots, k_m] &= \mathrm{DFT}(X_0[k_2, \ldots, k_m])[k_1] \\ &= \textstyle\sum_{j_1=1}^{n_1} x[j_1, k_2, \ldots, k_m] \cdot s_{k_1}[j_1] \\ &= \textstyle\sum_{j_1=1}^{n_1} \mathrm{DFT}_0(x[j_1, k_2, \ldots, k_m]) \cdot s_{k_1}[j_1] \end{aligned}$$

Nu geldt verder voor $t > 1$:

$$\begin{aligned} X_t[k_1, \ldots, k_m] &= DFT(X_{t-1}[k_1, \ldots, k_{t-1}, k_{t+1}, \ldots, k_m])[k_t] \\ &= \textstyle\sum_{j_t=1}^{n_t} X_{t-1}[k_1, \ldots, k_{t-1}, j_t, k_{t+1}, \ldots, k_m] \cdot s_{k_t}[j_t] \\ &= \textstyle\sum_{j_t=1}^{n_t} \sum_{j_{t-1}=1}^{n_{t-1}} \mathrm{DFT}_{t-2}(x[j_{t-1}, k_t, \ldots, k_m])[k_1, \ldots, k_{t-2}] \cdot s_{k_{t-1}}[j_{t-1}] \cdot s_{k_t}[j_t] \\ &= \textstyle\sum_{j_t=1}^{n_t} \mathrm{DFT}_{t-1}(x[j_t, k_{t+1}, \ldots, k_m])[k_1, \ldots, k_{t-1}] \cdot s_{k_t}[j_t] \end{aligned}$$

Waarbij we opeenvolgend onze de inductieaanname gebruiken en vervolgens de definitie van de MDFT. Hiermee geldt de relatie dus ook voor X_m, zodat dit algoritme de MDFT geeft. □

Een belangrijk gevolg is nu dat het algoritme in termen van de DFT direct vertaalt naar een multidimentionaal algoritme voor de FFT: beiden geven immers dezelfde output. Wanneer we overal de functie DFT vervangen door FFT dan geeft dit de Multidimentionale Fast Fourier Transform. De complexiteit van zo'n algoritme is $\mathcal{O}(\sum_{i=1}^{m} \log(n_i) \prod_{j=1}^{m} n_j)$ t.o.v. $\mathcal{O}(\sum_{i=1}^{m} n_i \prod_{j=1}^{m} n_j)$ voor de MDFT.

2.4 Compressie van Beeldmateriaal onder FFT

TODO: een klein introverhaaltje over compressie

Verwachte ontwikkeling van de fout

Het is zowel van praktisch nut als wiskundig interessant om het convergentie-gedrag te bepalen van de Fourier-transformatie. Met het convergentie gedrag voor continue functies kunnen we immers kwantificeren hoe goed een functie te benaderen is met een eindige subset van de Fourier-getransformeerde. We trachten daarom te bewijzen dat er een algebraïsch verband bestaat tussen de fout en de frequentie van de wavelets en bovendien een exponentieel verband tussen de fout en de 'gladheid' van de functie.

Voor dit bewijs zullen we het Lemma van Riemann-Lebesgue gebruiken:

Lemma (Riemann-Lebesque[2])**.** *Wanneer $g \in L_1(\mathbb{R})$, dan geldt*

$$G(z) = \left| \int_{-\infty}^{\infty} g(t) e^{-2\pi i z t} dt \right| \to 0 \text{ voor } z \to \infty.$$

Stelling (Daling van de coëfficiënten van de Fouriergetransformeerde)**.** *Laat $f \in L_2([a, b])$. Stel er is een n zó dat $f \in C^n$ en voor $0 \leq j \leq n$ geldt dat $f^{(j)}(a) = f^{(j)}(b)$. Dan geldt dat de k-de entry van de Fouriergetransformeerde $\hat{f}[k]$ in absolute waarde daalt met k volgens $\mathcal{O}(|k|^{-n})$.*

Bewijs. Laat $f \in L_2([a,b])$ zodat f voldoet aan de voorwaarden in de stelling voor een bepaalde n. De Fourier-getransformeerde van f wordt gegeven door:

$$\hat{f}[k] = \tfrac{1}{\sqrt{b-a}}\langle f, \phi_k \rangle.$$

De constante in deze vergelijking heeft geen invloed op de orde, dus richten we ons op het inproduct. We schrijven dit uit tot een integraal en voeren vervolgens, omdat f differentieerbaar is, partieële integratie uit.

$$\begin{aligned} |\langle f, \phi_k \rangle| = & \left| \int_a^b f(x) e^{-2\pi i k \frac{x-a}{b-a}} dx \right| = \left| \tfrac{b-a}{2\pi i k} \left[f(x) \cdot e^{-2\pi i k \frac{x-a}{b-a}} \right]_a^b \right| + \left| \tfrac{b-a}{2\pi i k} \right| \left| \int_0^1 f'(x) e^{-2\pi i k \frac{x-a}{b-a}} dx \right| \\ = & \quad |f(b) \cdot 1 - f(a) \cdot 1| + \tfrac{b-a}{2\pi |k|} \left| \int_a^b f'(x) e^{-2\pi i k \frac{x-a}{b-a}} dx \right| = \tfrac{b-a}{2\pi |k|} \left| \int_a^b f'(x) e^{-2\pi i k \frac{x-a}{b-a}} dx \right|. \end{aligned}$$

Hier hebben we gebruikt dat alle afgeleiden van 0 tot n periodiek zijn. Omdat f nu C^n is en de afgeleiden weer periodiek zijn kunnen we dit herhaald toepassen, we verkrijgen daarmee

$$\left| \int_a^b f(x) e^{-2\pi i k \frac{x-a}{b-a}} dx \right| = (\tfrac{2\pi}{b-a}|k|)^{-n} \left| \int_a^b f^{(n)}(x) e^{-2\pi i k \frac{x-a}{b-a}} dx \right|.$$

Vermenigvuldig beide kanten met $(\frac{2\pi}{b-a}|k|)^n$ om te vinden dat

$$(\tfrac{2\pi}{b-a}|k|)^n |\langle f, \phi_k \rangle| = \left| \int_a^b f^{(n)}(x) e^{-2\pi i k \frac{x-a}{b-a}} dx \right|.$$

We willen graag het lemma van Riemann-Lebesgue toepassen op de rechterkant. Merk daartoe op dat omdat $f \in C^n$, $f^{(n)}$ continu op $[a,b]$ dus neemt hier een maximum en een minimum aan. Daarmee is de integraal van $|f|$ begrensd en dus is $f \in L_1([a,b])$. Maar een functie die integreerbaar is op $[a,b]$ en daarbuiten nul, is integreerbaar op $\mathbb{R}$. Dus we mogen Riemann-Lebesgue gebruiken om te zien dat

$$(\tfrac{2\pi}{b-a}|k|)^n |\langle f, \phi_k \rangle| \to 0 \text{ als } |k| \to \infty.$$

Maar dit betekent precies dat $|\langle f, \phi_k \rangle| \in O((\frac{2\pi}{b-a}|k|)^{-n}) = O(|k|^{-n})$ □

De analyse van de coëfficiënten die we hier gegeven hebben vertaalt direct naar de fout die we krijgen bij reconstructie van een Fourier-getransformeerde functie wanneer we een gelimiteerde dataset gebruiken. Schrijf namelijk F voor de terug getransformeerde met een dataset waarbij k gelimiteerd is, en f voor de perfecte reconstructie. Dan hebben we de relatie:

$$\begin{aligned} ||f - F||_{L_2}{}^2 = & \left| \int_a^b \left(\sum_{k=1}^{\infty} \hat{f}[k] \phi_k(x) - \sum_{k=1}^{m-1} \hat{f}[k] \phi_k(x) \right)^2 \mathrm{d}x \right| \\ = & \left| \int_a^b \left(\sum_{k=m}^{\infty} \hat{f}[k] \phi_k(x) \right)^2 \mathrm{d}x \right| \\ \leq & \left| \sum_{k=m}^{\infty} \hat{f}^2[k] \cdot \int_a^b \phi_k^2(x) \mathrm{d}x \right| \leq \sum_{k=m}^{\infty} \left| \hat{f}[k] \right|^2. \end{aligned}$$

Omdat de orde van deze vergelijking in termen van m bepaald wordt door de grootste term kunnen we zeggen dat de orde van de fout gelijk is aan de orde van het grootste coëfficiënt:

$$||f - F||_{L_2([a,b])} \in \mathcal{O}\left(\left| \hat{f}[m] \right| \right) = \mathcal{O}(m^{-n})$$

De fout in discrete geval

TODO: discreet → continu voor grote dataset, maak een definitie van smooth op discrete functies

2.5 Implementatie van Fourier Transformatie in Python

TODO: introductie verhaaltje

Fast Fourier Transform Implementatie

TODO: even kijken of we dit zo moeten doen of dat we beter alles kunnen referen

De python code die gebruikt is om het FFT algoritme in te programmeren volgt precies het schema van Pseudocode zoals beschreven in een vorige sectie.

```
def FFT( xs ):
    N = len(xs)
    if N <= 1:                # randconditie
        return xs
    else:
        even = FFT(xs[0::2])  # voer FFT uit op even indices
        odd  = FFT(xs[1::2])  # voer FFT uit op oneven indices

        return [0.5*( even[k] + exp(-2j*pi*k/N)*odd[k] ) for k in range(N/2)] +
            [0.5*( even[k] - exp(-2j*pi*k/N)*odd[k] ) for k in range(N/2)]
```

FFT algoritme in Python, voert de pseudocode uit zoals in sectie (TODO)

Hierbij dient uitgelegd te worden dat de $[x \text{ for } y] + [z \text{ for } w]$ notatie twee lijsten construeert en achter elkaar zet. Dit algoritme werkt, zoals eerder beschreven enkel op signalen waarvan de lengte N een macht van 2 is. We hebben daarom signaalextensie toe moeten passen door middel van zero-padding om het programma ook op andersvormige signalen te laten werken.

```
def Zero_Padding( xs ):
    N_old = len(xs)
    N_new = 2**ceil(log(N_old,2))     # rond logaritme af voor kleinste tweemacht
    return [xs[k] if (k < N_old) else 0 for k in range(N_new)]
```

Zero-Padding algoritme in Python, voegt nullen toe tot een tweemacht is bereikt

Vervolgens hebben we deze code toegepast in twee dimensies. We maken hier gebruik van de definitie van het MFFT algoritme, dat grofweg zegt dat een meerdimensionaal algoritme kan

worden geconstrueerd door herhaald het 1-dimensionale geval toe te passen. Voor 2 dimensies in het bijzonder betekent dit dat we simpelweg FFT konden toepassen op rijen en kolommen.

```
def FFT_2D( xss ):
    xss = map(FFT, xss)   # voer FFT uit op rijen

    xss_t = transpose(xss)  # verwissel rijen met kolommen
    xss_t = map(FFT, xss_t) # voer FFT uit op kolommen
    xss = transpose(xss_t)  # maak verwisseling ongedaan

    return xss
```

2-Dimensionaal FFT algoritme

Hier is de python-functie *map* gebruikt die ruwweg gedefinieerd is als $\text{map}(f, x) = [f(y) \text{ for } y \text{ in } x]$ Met het oog op duidelijkheid is hier de Zero-Padding fase weggelaten, dit algoritme verwacht nu nog een $2^n \times 2^m$ matrix. Dit is echter gemakkelijk te implementeren door Zero_Padding toe te passen op rijen en kolommen.

TODO: meer implementatie specifieke details includen

Hoofdstuk 3

Wavelets

De Fouriertransformatie bestaat al honderden jaren en is een grote speler geworden in de *signal processing*. Een groot nadeel van deze transformatie is dat zij slecht reageert op discontinue signalen door de drager van de basisfuncties.

In de loop van de vorige eeuw is een nieuwe transformatie ontstaan met een eigenschap die de Fouriertransformatie nooit kende. Deze noemt men nu ook wel de Wavelettransformatie.

Definitie. *Een wavelet is simpelweg een functie* $\psi : \mathbb{R} \to \mathbb{R}$ *die voldoet aan*

$$z \int_{-\infty}^{\infty} \psi(t)dt = 0.$$

Met deze functie ψ *kunnen we een familie functies* $\psi_{u,s}$ *bouwen door middel van schaling en translatie:*

$$\psi_{u,s}(t) := \frac{1}{\sqrt{s}} \psi \left(\frac{t-u}{s} \right).$$

Deze familie geeft aanleiding tot een Wavelettransformatie W_f van f:

$$W_f(u,s) = \int_{-\infty}^{\infty} f(t)\psi^*_{u,s}(t)dt.$$

Het is nu mogelijk om wavelets te construeren die met deze schaling en translatie een basis voor de $L_2(\mathbb{R})$ vormen. Over het algemeen kijken we dan naar

$$\Psi := \left\{ \psi_{j,n}(t) = \frac{1}{\sqrt{2^j}} \psi \left(\frac{t - 2^j n}{2^j} \right) : (j,n) \in \mathbb{Z}^2 \right\}.$$

De kunst is nu om de basiselementen loodrecht op elkaar te laten staan, zodat er een orthogonale (en dus een orthonormale) basis gevormd wordt.

Figuurtje van voorbeeld TODODOO.

Gevolg. We kunnen een functie f in $L^2(\mathbb{R})$ schrijven in deze basis:

$$f(t) = \sum_{j=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} \langle f, \psi_{j,n} \rangle \psi_{j,n}(t),$$

waarbij $\langle \cdot, \cdot \rangle$ het standaardinproduct op de $L_2(\mathbb{R})$ aangeeft.

Het grote nadeel van de Fouriertransformatie maakt compressie van discrete signalen moeilijk. Veel van deze wavelets worden nu zó geconstrueerd dat dit probleem (deels) verholpen wordt. We zijn namelijk op zoek naar een wavelet die een eindige drager heeft. Het blijkt dat deze bestaat en dat er zelfs een hele grote verzameling wavelets is, elk met eigen gewilde eigenschappen.

Omdat wij naar de toepassing van wavelets binnen de beeldcompressie bekijken, zijn we natuurlijk vooral geïnteresseerd in het discrete geval. We kijken dus naar de benadering van f op een TODO. Dit geeft aanleiding tot een rij geneste ruimtes die uiteindelijk naar de $L_2(\mathbb{R})$ toe gaat

$$\{0\} \ldots \subset V_{j+1} \subset V_j \subset V_{j-1} \subset \ldots \subset L_2(\mathbb{R}) \tag{3.1}$$

genaamd een multiresolutie.

Definitie. *Een rij geneste ruimtes $\{V_j : j \in \mathbb{Z}\}$ zoals in 3.1 heet een multiresolutie wanneer voldaan wordt aan de volgende eigenschappen:*

$$\forall j, k : f(t) \in V_j \implies f(t - 2^j k) \in V_j, \tag{3.2}$$

$$\forall j : V_{j+1} \subset V_j, \tag{3.3}$$

$$\forall j : f(t) \in V_j \iff f(t/2) \in V_{j+1}, \tag{3.4}$$

$$\bigcap_{j=-\infty}^{\infty} V_j = \lim_{j \to \infty} V_j = \{0\}, \tag{3.5}$$

$$\bigcup_{j=-\infty}^{\infty} V_j = \lim_{j \to -\infty} V_j = L_2(\mathbb{R}), \tag{3.6}$$

$$TODO: \text{ Er is } \theta \text{ zo dat } \{\theta(t-n) : n \in \mathbb{Z}\} \text{ een Rieszbasis voor } V_0 \text{ is.} \tag{3.7}$$

Voorbeeld. We bekijken een multiresolutie van stuksgewijs constante functies. De ruimte V_j wordt hiermee de verzameling van alle $g(t) \in L_2(\mathbb{R})$ die constant zijn voor $t \in [n2^j, (n+1)2^j)$ met $n \in \mathbb{Z}$. De basisfunctie θ voor V_0 wordt in dit geval $\theta(t) = 1_{[0,1)}$.

3.1 Schalingsfuncties

Gegeven zo'n Rieszbasis voor V_0 willen we graag een orthonormale basis voor V_j construeren.

Stelling ([5] T7.1)**.** *Laat $\{V_j\}$ een multiresolutie en laat ϕ de schalingsfunctie waarvan de Fouriergetransformeerde voldoet aan*

$$\hat{\phi}(\omega) = \frac{\hat{\theta}(\omega)}{\left(\sum_{k=-\infty}^{\infty} |\hat{\theta}(\omega + 2k\pi)|^2\right)^{1/2}}.$$

Laat verder

$$\phi_{j,n}(t) := \frac{1}{\sqrt{2^j}}\phi\left(\frac{t-n}{2^j}\right).$$

Dan is $\{\phi_{j,n} : n \in \mathbb{Z}\}$ *een orthonormale basis voor* V_j.

We zullen ons verder niet bezighouden met de Fouriergetransformeerde en alleen het gevolg (de orthonormale basis) gebruiken. Hij staat er bij om duidelijk te maken dat deze schalingsfunctie expliciet gevonden kan worden.

Benadering

De orthogonale projectie van f op V_j is, zoals we weten, de beste benadering van f in V_j. Deze is nu te vinden door

$$P_{V_j} f = \sum_{n=-\infty}^{\infty} \langle f, \phi_{j,n}\rangle \phi_{j,n}.$$

De coëfficiënten $a_j[n] = \langle f, \phi_{j,n}\rangle$ geven ons op deze manier een discrete benadering van f op resolutie 2^{-j}.

Voorbeeld. De multiresolutie van stuksgewijs constante functies op de intervallen $[2^j n, 2^j(n+1))$ met $n \in \mathbb{Z}$ heeft als eigenschap dat de Rieszbasis ook direct orthonormaal is. Gevolg is dat $\phi = \theta = 1_{[0,1]}$.

3.2 Filters

Wanneer we een schalingsfunctie ϕ definiëren (en dus een V_0), ligt de hele multiresolutie eigenlijk vast. We zullen daarom deze schalingsfunctie nader onderzoeken.

Per definitie van de multiresolutie weten we dat $V_j \subset V_{j-1}$. In het bijzonder geldt dat $2^{-1/2}\phi(t/2) \in V_1 \subset V_0$ en omdat $\{\phi(t-n) : n \in \mathbb{Z}\}$ een orthonormale basis voor V_0 is, kunnen we $2^{-1/2}\phi(t/2)$ nu schrijven als

$$\frac{1}{\sqrt{2}}\phi\left(\frac{t}{2}\right) = \sum_{n=-\infty}^{\infty} \left\langle \frac{1}{\sqrt{2}}\phi\left(\frac{t}{2}\right), \phi(t-n)\right\rangle \phi(t-n).$$

Definitie. *Deze inproducten hebben een speciale naam, want de rij* $\{h[n] : n \in \mathbb{Z}\}$ *met*

$$h[n] := \left\langle \frac{1}{\sqrt{2}}\phi\left(\frac{t}{2}\right), \phi(t-n)\right\rangle$$

wordt nu ook wel de filter *van* ϕ *genoemd.*

Stelling ([5][T7.2])**.** *Laat* $\phi \in L^2(\mathbb{R})$ *een schalingsfunctie die ook integreerbaar is. Dan ligt de multiresolutie vast.*

Andersom, als $h[n]$ een filter is zodat $\hat{h}(\omega)$ periodiek 2π is en continu differentieerbaar in een omgeving van $\omega = 0$ en als daarnaast geldt

$$\begin{aligned} \forall \omega \in \mathbb{R} : |\hat{h}(\omega)|^2 + |\hat{h}(\omega + \pi)|^2 &= 2, \\ \hat{h}(0) &= \sqrt{2}, \\ \inf_{\omega \in [-\pi/2, \pi, 2]} |\hat{h}(\omega)| &> 0, \end{aligned}$$

dan is de functie ϕ waarvan de Fouriergetransformeerde voldoet aan

$$\hat{\phi}(\omega) = \prod_{p=1}^{\infty} \frac{\hat{h}(2^{-p}\omega)}{\sqrt{2}}$$

een schalingsfunctie in $L^2(\mathbb{R})$.

Wederom zullen we enkel de gevolgen gebruiken: namelijk dat de multiresolutie vast ligt met een goede keuze van ϕ, en dat voor een goed gekozen $h[n]$, ϕ ook vast ligt.

Voorbeeld. Bekijk weer het geval $\phi(t) = 1_{[0,1)}$. Dan vinden we dat

$$h[n] = \left\langle \frac{1}{\sqrt{2}} \phi\left(\frac{t}{2}\right), \phi(t-n) \right\rangle = \begin{cases} \frac{1}{\sqrt{2}} & \text{als } n \in \{0, 1\} \\ 0 & \text{anders.} \end{cases}$$

3.3 Terugkeer van de wavelet

We weten dat V_j bevat is in V_{j-1}. Laat nu W_j het orthogonale complement van V_j in V_{j-1}:

$$W_j \oplus V_j = V_{j-1}.$$

De projectie van f op V_{j-1} kan dus geschreven worden als som van projecties:

$$P_{V_{j-1}} f = P_{V_j} f + P_{W_j} f.$$

Omdat $V_j \subset V_{j-1}$ is alle informatie over f die beschikbaar is in V_j, ook beschikbaar in V_{j-1}. Ook is het goed mogelijk dat door deze grovere benadering, informatie zoek gaat. Deze 'details' worden op die manier zichtbaar in $P_{W_j} f$.

Het kan bewezen worden[5][T7.3] dat, gegeven een schalingsfunctie ϕ (en daarmee een filter h) er een functie ψ bestaat zo dat

$$\left\{ \psi_{j,n}(t) := \frac{1}{\sqrt{2}} \psi\left(\frac{t - 2^j n}{2^j}\right) : n \in \mathbb{Z} \right\}$$

een orthonormale basis is voor W_j en $\{\psi_{j,n} : (j,n) \in \mathbb{Z}^2\}$ een basis voor $L_2(\mathbb{R})$. Deze functie is zo een orthogonale wavelet, omdat $W_j \perp V_j$. Omdat nu $W_j \subset V_{j-1}$ en dus in het bijzonder

$2^{-1/2}\psi(t/2) \in W_1 \subset V_0$ en omdat $\{\phi(t-n) : n \in \mathbb{Z}\}$ een orthonormale basis is voor V_0, kunnen we ook $2^{-1/2}\psi(t/2)$ in termen schrijven als:

$$\frac{1}{\sqrt{2}}\psi\left(\frac{t}{2}\right) = \sum_{n=-\infty}^{\infty} \left\langle \frac{1}{\sqrt{2}}\psi\left(\frac{t}{2}\right), \phi(t-n) \right\rangle \phi(t-n).$$

Ook deze inproducten hebben een speciale naam: de rij $g[n]$ met

$$g[n] := \left\langle \frac{1}{\sqrt{2}}\psi\left(\frac{t}{2}\right), \phi(t-n) \right\rangle$$

wordt nu ook wel de filter van ψ genoemd. De twee filters zijn gerelateerd aan elkaar volgens de vergelijking

$$g[n] = (-1)^{1-n} h[1-n].$$

Zoals nu wel duidelijk geworden is, wordt met een filter h (die voldoet aan bepaalde eigenschappen: zie stelling 3.2) een schalingsfunctie ϕ en een filter g met waveletfunctie ψ geconstrueerd.

Voorbeeld. We keren nog een laatste keer terug naar het voorbeeld waarin $\phi(t) = 1_{[0,1)}$. We vinden met de gelijkheden uit voorgaande paragrafen dat

$$\frac{1}{\sqrt{2}}\psi\left(\frac{t}{2}\right) = \sum_{n=-\infty}^{\infty} (-1)^{1-n} h[1-n]\phi(t-n),$$

en omdat $h[0] = h[1] = 2^{-1/2}, h[n] = 0$ voor $n \in \mathbb{Z} \setminus \{0,1\}$ zoals we eerder vonden, herschrijft dit tot

$$\frac{1}{\sqrt{2}}\psi\left(\frac{t}{2}\right) = \frac{1}{\sqrt{2}}\left(\phi(t-1) - \phi(t)\right)$$

met als gevolg dat

$$\psi(t) = \begin{cases} -1 & \text{als } t \in [0, 1/2) \\ 1 & \text{als } t \in [1/2, 1) \\ 0 & \text{anders.} \end{cases}$$

Deze wavelet ψ wordt ook wel de Haarwavelet genoemd en is uitgevonden voor Alfred Haar in 1909, hoewel het onderzoeksgebied van de wavelets toen nog niet bestond. In het vervolg zullen we nog verdere aandacht aan deze wavelet besteden.

3.4 Het kiezen van een wavelet

Bij het kiezen of vinden van een wavelet is men over het algemeen op zoek naar bepaalde eigenschappen. Voor compressie zijn we op zoek naar een wavelet die voor bepaalde klassen functies een klein aantal grote coëfficiënten en een groot aantal kleine teweeg brengt: een soort concentratie van de coëfficiënten. Dit wordt vooral bepaald door drie factoren: gladheid van f (waar we niks aan kunnen doen), de grootte van de drager (welke hierna aan bod komt) en de zogenaamde orde van de wavelet.

Definitie. *Wanneer de waveletfunctie loodrecht staat ($\langle \psi, q \rangle = 0$) op alle polynomen van graad $p-1$ of lager, spreken we van een wavelet van orde p. Dit komt overeen met te zeggen dat*

$$\int_{-\infty}^{\infty} x^k \psi(x) dx = 0 \text{ voor } k \in \{0, \dots p-1\}.$$

Gevolg. Gevolg van deze eigenschap is dat we van de functie f elk polynoom van graad $p-1$ af mogen trekken zonder een verschil in inproduct:

$$\langle f, \psi_{j,n} \rangle = \langle f - q, \psi_{j,n} \rangle \text{ voor } q \text{ een polynoom van graad } p-1.$$

Intuïtief is deze eigenschap natuurlijk gewild: we winnen immers een heel stel keuzevrijheden. We zullen dit argument in een volgende sectie formaliseren.

Eerder spraken we het verlangen uit om een wavelet met eindige drager te vinden zodat discontinuïteiten alleen lokaal zichtbaar zijn. We zullen hier de dragers van $h[n], \psi$ en ϕ aan elkaar relateren.

Compacte drager

Stelling ([5] P7.2)**.** *De volgende relaties gelden voor de dragers.*

1. *De schalingsfunctie ϕ heeft een compacte drager dan en slechts dan als $h[n]$ een compacte drager heeft, en deze zijn hetzelfde.*
2. *Als de drager van ϕ gelijk is aan $[N_1, N_2]$ dan is de drager van ψ gelijk aan $[(N_1 - N_2 + 1)/2, (N_2 - N_1 + 1)/2]$.*

Bewijs 1. Als ϕ een compacte drager heeft dan $h[n]$ ook: we weten dat

$$h[n] = \left\langle \frac{1}{\sqrt{2}} \phi\left(\frac{t}{2}\right), \phi(t-n) \right\rangle,$$

er kunnen maar eindig veel n ongelijk nul zijn. Omgekeerd, als $h[n] \neq 0$ voor eindig veel n, dan zien we

$$\frac{1}{\sqrt{2}} \phi\left(\frac{t}{2}\right) = \sum_{n=-\infty}^{\infty} h[n] \phi(t-n) \tag{3.8}$$

TODO: pag 966 van Daubechies "orthonormal bases of compactly supported wavelets".

Om deze dragers gelijk te krijgen, stel dat de drager van $h[n]$ gelijk $[N_1, N_2]$ is, en die van ϕ $[K_1, K_2]$. De drager van $\phi(t/2)$ is $[2K_1, 2K_2]$ en de drager van de rechterzijde van 3.8 is $[N_1 + K_1, N_2 + K_2]$. We concluderen dat $K_1 = N_1$ en $K_2 = N_2$. □

Bewijs 2. Kijk nu naar

$$\frac{1}{\sqrt{2}}\psi\left(\frac{t}{2}\right) = \sum_{n=-\infty}^{\infty} g[n]\phi(t-n) = \sum_{n=-\infty}^{\infty} (-1)^{1-n}h[-1-n]\phi(t-n).$$

Met de informatie uit het begin van de stelling kunnen we de drager van de rechterkant vinden: $[N_1 - N_2 + 1, N_2 - N_1 + 1]$. De functie $\psi(t/2)$ is nu precies een dilatie met factor twee dus de drager van $\psi(t)$ moet wel gelijk zijn aan $[(N_1 - N_2 + 1)/2, (N_2 - N_1 + 1)/2]$. □

Daubechieswavelets

Hoewel de constructie van de Daubechieswavelet buiten het spectrum van dit artikel valt,[1] willen we toch een kort licht schijnen op deze speciale familie van wavelets. Deze worden gemaakt met de noties eerder, namelijk dat we de drager willen minimaliseren maar de orde maximaliseren. Daubechies heeft bewezen [TODO ref] dat een filter h met orde p, minimaal een drager van lengte $2p$ moet hebben.

De Daubechieswavelet van orde p nu, heeft precies een filter van lengte $2p$. In het bijzonder is de Haarwavelet de eerste in de familie van Daubechieswavelets.

Wij hebben in het praktische deel van ons project aandacht besteed aan de zogenaamde Daubechies-2 wavelet die haar naam ontleent aan het feit dat zij van orde 2 is.

TODO picca van Daubechies wavelets

3.5 Fast Wavelet Transform

TODO schrijven dit probeer dit er in te stoppen (als dat niet al duidelijk wordt)?

$$V_{-J} = V_0 \oplus W_0 \oplus \cdots \oplus W_{1-J}$$

$\mathbb{N}$***oem ook*** $\mathbb{C}$***onvolutie***

$$a_{j+1}[n] = \sum_{p=-\infty}^{\infty} h[p-2n]a_j[p] = a_j \star \bar{h}[2n] \tag{3.9}$$

$$d_{j+1}[n] = \sum_{p=-\infty}^{\infty} g[p-2n]a_j[p] = a_j \star \bar{g}[2n] \tag{3.10}$$

[1] Voor een goede beschrijving van deze constructie, zie [5].

Onze implementatie

3.6 Analyse van de Wavelettransformatie

Met de theoretische beschouwing van wavelets en de Fast Wavelet Transform achter de rug, kunnen we wat verder kijken naar practische obstakels.

Eindige signalen

Een van de eerste aannames die we tot nu toe steeds maakten is die van de oneindige signalen. Wanneer echter de functie f een compacte drager heeft, worden een aantal zaken wat lastiger. Neem als eerste aan dat de drager van f gewoon $[0, 1]$ is.[2] In dit geval zou het kunnen dat de waveletfuncties met een drager die $t = 0$ of $t = 1$ doorsnijdt, niet meer van een bepaalde orde is. Er zijn in de literatuur oplossingen voor dit probleem gevonden. Hier zullen wij verder niet op in gaan.

Wanneer f leeft in $L^2([0, 1])$, kunnen we alle V_j met $j > 0$ buiten beschouwing laten omdat de resolutie in deze ruimte te laag is om nog interessante informatie over f te bieden. We hebben op deze manier een rij geneste ruimtes

$$V_0 \subset V_{-1} \subset \dots.$$

Wanneer we nu een benadering van f maken op resolutie 2^{-J} (door bijvoorbeeld een Fast Wavelet Transform), bekijken we

$$V_0 \subset V_{-1} \subset \dots \subset V_{-J}.$$

Een discreet signaal van lengte 2^{-J} kan zo perfect 'benaderd' worden in een waveletbasis op resolutie 2^{-J} en dit is precies waarom de Fast Wavelet Transform zo veel gebruikt wordt bij het analyseren van discrete signalen.

Signaaluitbreiding

Een probleem waar we in het geval van eindige signalen nog meer mee te maken krijgen is dat de algoritme niet goed omgaat met de randen. De convolutie moet nu ineens *buiten het definitiegebied* van het signaal 'kijken'. Eerder in sectie [TODO] hebben we al gezien hoe signalen naar een tweemacht uitgebreid kunnen worden. Precies dezelfde methoden kunnen gebruikt worden om het signaal nog verder uit te breiden.

Om niet te veel tijd te verliezen met het ondersteunen van meerdere mogelijkheden hebben wij er voor gekozen om periodic padding op alle signalen toe te passen. Dit omdat de zogenaamde *circulaire convolutie* ingebouwd zit in de bibliotheek die wij gebruikt hebben.

[2] Door translatie en dilatie kan elk signaal met compacte basis omgevormd worden tot een signaal met drager in $[0, 1]$. We verliezen hier dus geen algemeenheid.

Complexiteit van de algoritme

Als de lengte van de filter h gelijk is aan K, en de lengte van het originele signaal a_L gelijk is aan $N = 2^{-L}$, kunnen we voor $j \in \{L, \ldots, 0\}$ zien dat a_j en d_j beide 2^{-j} elementen bevatten. Nu kunnen a_{j+1} en d_{j+1} gemaakt worden door $2^{-j}K$ operaties zodat elke stap van de algoritme $2^{-j} \cdot K$ operaties kost. Dan kost het hele algoritme

$$\sum_{j=L}^{0} 2^{-j} \cdot K = K \sum_{j=L}^{0} 2^{-j} = K \cdot (2^{1-L} - 1) < 2 \cdot K 2^{-L} = 2KN$$

operaties. Dus deze DWT is een $\mathcal{O}(KN)$ algoritme. Ook de complexiteit van de inverse wordt op dezelfde manier van orde KN.

3.7 Meer dimensies: de Mallatdecompositie

Met een orthonormale waveletbasis $\{\psi_{j,n} : (j,n) \in \mathbb{Z}^2\}$ van $L_2(\mathbb{R})$ volgt een natuurlijke voortzetting naar twee dimensies door

$$\{\psi_{j_1,n_1}(x_1)\psi_{j_2,n_2}(x_2) : (j_i, n_i) \in \mathbb{Z}^4\},$$

welke een orthonormale basis voor $L_2(\mathbb{R}^2)$ is. We zien direct dat we op de x_1-as met resolutie 2^{-j_1} kijken terwijl de x_2-as resolutie 2^{-j_2} kent. Mallat vond dit iets om te vermijden [5][§7.7] en legt in zijn analyse dan ook de eis $j_1 = j_2 =: j$ op. Wij zullen in het vervolg kijken naar het zogenaamde Tensorproduct, wat de eis $j_1 = j_2$ *niet* oplegt.

Zoals in 1 dimensie is de notie van 'resolutie' geformaliseerd in het begrip multiresolutie. De definitie van deze multiresolutie is wederom een natuurlijke voortzetting van het eendimensionale geval. Op deze manier bekijken we een ruimte $V_j^2 = V_j \otimes V_j$. In [5] §7.7 wordt nu bewezen dat, gegeven een orthonormale basis $\{\phi_{j,m} : m \in \mathbb{Z}\}$ voor V_j, de verzameling

$$\{\phi_{j,n}^2 := \phi_{j,n_1} \otimes \phi_{j,n_2} : n \in \mathbb{Z}^2\}$$

een basis voor V_j^2 is.

Neem als voorbeeld weer V_j de ruimte van stuksgewijs constante functies op het interval $[2^j m, 2^j(m+1))$ voor $m \in \mathbb{Z}$. We vinden voor V_j^2 nu de ruimte van vierkantsgewijs constante functies op vierkanten $[2^j n_1, 2^j(n_1+1)) \times [2^j n_2, 2^j(n_2+1))$. De tweedimensionale schalingsfunctie wordt op die manier

$$\phi^2(x_1, x_2) = \phi(x_1)\phi(x_2) = \begin{cases} 1 & \text{als } x_1 \in [0,1) \text{ en } x_2 \in [0,1) \\ 0 & \text{anders.} \end{cases}$$

Tweedimensionale Waveletfuncties

We weten dat V_j^2 bevat is in V_{j-1}^2. We bekijken het orthogonale complement $\boldsymbol{U}_j \perp V_j^2$:

$$V_j^2 \oplus U_j = V_{j-1}^2.$$

Om nu een orthogonale waveletbasis voor $\boldsymbol{U}_j$ en (dus) $L^2(\mathbb{R}^2)$ te vinden, gaan we als volgt te werk.

Stelling ([5] T7.24)**.** *Laat ϕ een schalingsfunctie en ψ de bijhorende wavelet die en basis voor de $L^2(\mathbb{R})$ voortbrengt. Maak drie wavelets*

$$\psi^1(x) = \phi(x_1)\psi(x_2) \quad \psi^2(x) = \psi(x_1)\phi(x_2) \quad \psi^3(x) = \psi(x_1)\psi(x_2)$$

en laat voor $k \in \{1,2,3\}$ nu

$$\psi_{j,n}^k(x) = \frac{1}{2^j}\psi^k\left(\frac{x_1 - 2^j n_1}{2^j}, \frac{x_2 - 2^j n_2}{2^j}\right).$$

Dan is $\{\psi_{j,n}^1, \psi_{j,n}^2, \psi_{j,n}^3 : n \in \mathbb{Z}^2\}$ een basis voor $\boldsymbol{U}_j$ en $\{\psi_{j,n}^1, \psi_{j,n}^2, \psi_{j,n}^3 : n \in \mathbb{Z}^2, j \in \mathbb{Z}\}$ een basis voor $L^2(\mathbb{R}^2)$.

Bewijs. We weten

$$V_{j-1}^2 = V_j^2 \oplus \boldsymbol{U}_j \implies V_{j-1} \otimes V_{j-1} = (V_j \otimes V_j) \oplus \boldsymbol{U}_j.$$

Vul nu $V_{j-1} = V_j \oplus W_j$ in om te vinden

$$(V_j \oplus W_j) \otimes (V_j \oplus W_j) = (V_j \otimes V_j) \oplus \boldsymbol{U}_j$$

$$\implies (V_j \otimes V_j) \oplus (V_j \otimes W_j) \oplus (W_j \otimes V_j) \oplus (W_j \otimes W_j) = (V_j \otimes V_j) \oplus \boldsymbol{U}_j$$

$$\implies (V_j \otimes W_j) \oplus (W_j \otimes V_j) \oplus (W_j \otimes W_j) = \boldsymbol{U}_j.$$

Nu is het duidelijk dat $\{\psi_{j,n}^1, \psi_{j,n}^2, \psi_{j,n}^3 : n \in \mathbb{Z}^2\}$ een basis is voor $\boldsymbol{U}_j$. Omdat nu

$$L^2(\mathbb{R}^2) = \bigoplus_{j=-\infty}^{\infty} \boldsymbol{U}_j$$

is $\{\psi_{j,n}^1, \psi_{j,n}^2, \psi_{j,n}^3 : n \in \mathbb{Z}^2, j \in \mathbb{Z}\}$ een basis voor $L^2(\mathbb{R}^2)$. □

Met deze kennis in handen zullen we het tweedimensionale geval van de Fast Wavelet Transform formuleren.

Algoritme (Tweedimensionale Fast Wavelet Transform)**.** *Zoals in het eendimensionale geval, laat j vast. Dan:*

$$a_j[n] := \langle f, \phi_{j,n}^2 \rangle \quad d_j^k[n] := \langle f, \psi_{j,n}^k \rangle, k \in \{1,2,3\}.$$

Vervolgens, gebruikmakend van vergelijkingen 3.9 *en* 3.10*, zien we dat (TODO):*

$$a_{j+1}[n] = a_j \star (\bar{h} \otimes \bar{h})[2n];$$
$$d^1_{j+1}[n] = a_j \star (\bar{h} \otimes \bar{g})[2n];$$
$$d^2_{j+1}[n] = a_j \star (\bar{g} \otimes \bar{h})[2n];$$
$$d^3_{j+1}[n] = a_j \star (\bar{g} \otimes \bar{g})[2n],$$

waarbij

$$g \star (x \otimes y)[n_1, n_2] := \sum_{p_1=-\infty}^{\infty} x[n_1 - p_1] \sum_{p_2=-\infty}^{\infty} y[n_2 - p_2] g[p_1, p_2]$$

een tweedimensionale convolutie definiëert.

Andersom geldt dat

$$a_j[n] = \breve{a}_{j+1} \star (h \otimes h)[n] + \breve{d}^1_{j+1} \star (h \otimes g)[n] + \breve{d}^2_{j+1} \star (g \otimes h)[n] + \breve{d}^3_{j+1} \star (g \otimes g)[n]$$

TODO: ogier schrijf jij dit af?

Wat we steeds doen is de ruimte V_j^2 schrijven als V_{j+1}^2 met daarbij een aantal orthogonale complementen. Wanneer dit algoritme recursief toegepast wordt, schrijven we V_{j+1}^2 ook weer als V_{j+2}^2 met orthogonale complementen en zo voort. Dit levert dat

$$\mathbf{\Phi} = \left\{ \psi^k_{j+1,\cdot}, \psi^k_{j+2,\cdot}, \ldots \psi^k_{L,\cdot}, \phi^2_{L,\cdot} : k \in \{1, 2, 3\} \right\}$$

een basis is voor V_j^2. TODO plaatje. Dit is echter niet de enige basis die we kunnen bouwen voor V_j^2 wanneer we recursief tot een niveau L kijken.

Meer dan twee dimensies

Het n-dimensionale geval van de Mallatdecompositie is nog TODOOO

Eindige signalen

Concreet zullen we echter niet gebruik maken van signalen die leven in $L^2(\mathbb{R}^n)$. We zullen eerder op zoek zijn naar de Wavelettransformatie van een signaal met compacte drager. Neem dus aan dat f leeft in $L^2([0,1]^n)$. Dit n-dimensionale eenheidsinterval wordt ook wel met $\square$ aangegeven. Omdat $\square \subset \mathbb{R}^n$, kunnen we een waveletbasis voor $L^2(\mathbb{R}^n)$ ook als basis nemen voor $L^2(\square)$. Maar eigenlijk is dit iets te veel (gezien het feit dat veel basisfuncties hun drager geheel buiten het interval zullen hebben). Daarom wordt de verzameling van indices $\lambda := (j, n)$ van wavelets die drager doorsnijden met $[0, 1]$ ook wel ∇ genoemd. Dus $\Psi = \{\psi_\lambda : \lambda \in \nabla\}$ wordt nu een basis voor $L^2([0, 1])$. Definiëer $|\lambda| = |(j, n)| = j$ als het niveau. Verder zullen

we vanaf nu aannemen dat ψ een compacte drager heeft (zoals we in de praktijk altijd willen) en van orde p is.

TODO: definitie H^d -¿ Sobolevruimte.

Analyse van de fout van een lineaire benadering

Bij compressie is men op zoek naar een manier om stukjes data weg te kunnen gooien of te schrijven op zo'n manier dat het minder ruimte inneemt. Wij zijn in het bijzonder geïnteresseerd naar *hoe dichtbij* we bij perfecte reconstructie zitten wanneer we een vooraf bepaald aantal data opslaan. Er is al uitgebreid onderzoek gedaan naar hoe dit werkt bij de waveletbasis en een aantal resultaten hiervan zullen we opnemen in ons verslag.

Laat f een functie in $L^2(\square)$ met aftelbare basis $\mathcal{B} = \{g_m\}$. Dan valt f in deze basis te schrijven als

$$f = \sum_{m=0}^{\infty} \langle f, g_m \rangle g_m.$$

Lemma. *Als nu een functie f in $L^2(\square)$ geschreven wordt in $\mathcal{B}$ dan geldt*

$$\|f\|^2 = \sum_{m=0}^{\infty} |\langle f, g_m \rangle|^2.$$

Bewijs. We hebben te maken met een Hilbertruimte dus

$$\|f\|^2 = \langle f, f \rangle = \left\langle \sum_{m=0}^{\infty} \langle f, g_m \rangle g_m, \sum_{n=0}^{\infty} \langle f, g_n \rangle g_n \right\rangle = \sum_{m=0}^{\infty} \sum_{n=0}^{\infty} \langle \langle f, g_m \rangle g_m, \langle f, g_n \rangle g_n \rangle$$

$$= \sum_{m=0}^{\infty} \sum_{n=0}^{\infty} \langle f, g_m \rangle \overline{\langle f, g_n \rangle} \langle g_m, g_n \rangle = \sum_{m=0}^{\infty} \sum_{n=0}^{\infty} \langle f, g_m \rangle \overline{\langle f, g_n \rangle} \delta_{m,n}$$

$$= \sum_{m=0}^{\infty} \langle f, g_m \rangle \overline{\langle f, g_m \rangle} = \sum_{m=0}^{\infty} |\langle f, g_m \rangle|^2.$$

□

Wanneer we nu niet de hele basis, maar zeg alleen de eerste N elementen pakken, krijgen we een verzameling $\mathcal{B}_N \subset \mathcal{B}$ zodat

$$f_{\mathcal{B}_N} := \sum_{m=0}^{N-1} \langle f, g_m \rangle g_m.$$

In het bijzonder zijn we nu op zoek naar de *fout* $\|f - f_{\mathcal{B}_N}\|$:

$$\|f - f_{\mathcal{B}_N}\|^2 = \left\| \sum_{m=0}^{\infty} \langle f, g_m \rangle g_m - \sum_{m=0}^{N-1} \langle f, g_m \rangle g_m \right\|^2 = \left\| \sum_{m=N}^{\infty} \langle f, g_m \rangle g_m \right\|^2 = \sum_{m=N}^{\infty} |\langle f, g_m \rangle|^2.$$

Duidelijk moge zijn dat voor $N \to \infty$, $\|f - f_{\mathcal{B}_N}\|^2 \to 0$.

Fout van de Mallatdecompositie

Wij zijn op het moment geïnteresseerd in de Mallat-waveletbasis $\mathbf{\Phi}$. Deze is ook duidelijk aftelbaar dus bovenstaande eigenschap geldt.

Definiëer $J_M := \{l \in \mathbb{N}^n : \|l\|_\infty \leq M\}$. Laat $\mathbf{\Phi}_M := \{\boldsymbol{\psi}_{\boldsymbol{\lambda}} \in \mathbf{\Phi} : |\boldsymbol{\lambda}| \in J_M\}$ met $|\boldsymbol{\lambda}| = (|\lambda_1, \ldots, |\lambda_n|)$ de verzameling basisfuncties tot een niveau M.

Stelling (Fout van Mallatdecompositie)**.** *Wanneer $f \in H^p(\square)$, zal de fout $\|f - f_{\mathbf{\Phi}_M}\|$ bij een Mallatdecompositie met de basisfuncties tot niveau M hoogstens proportioneel met $(\#\mathbf{\Phi}_M)^{-p/n}$ zijn.*

Bewijs. We maken gebruik van de zogenaamde Jacksonongelijkheid [6] die zegt dat

$$\inf_{q \in \mathbb{P}_{p-1}} ||f - q||_{L_2(\square)} \simeq 2^{-jp} ||f||_{H^p(\square)}$$

wanneer $f \in H^p(\square)$.

Voor elke dimensie zitten er $\mathcal{O}(2^M)$ basisfuncties in $\{\psi_\lambda : |\lambda| \leq M\}$. Er zijn n dimensies dus $2^{Mn} = N(= \#\mathbf{\Phi})$ basisfuncties in totaal.

Bekijk de fout:

$$\|f - f_{\mathbf{\Phi}_M}\|^2_{L_2(\square)} = \sum_{\boldsymbol{\psi} \in \mathbf{\Phi} \setminus \mathbf{\Phi}_M} |\langle f, \boldsymbol{\psi} \rangle|^2 \simeq \sum_{|\boldsymbol{\lambda}| > M} 2^{-|\boldsymbol{\lambda}|p} \simeq 2^{-Mp},$$

waarbij het laatste isteken voortkomt uit

$$\sum_{k=M+1}^{\infty} 2^{-kp} = \frac{2^{-Mp}}{2^p - 1}$$

en de notie dat p constant is voor een keuze van de wavelet. De fout is nu $2^{-Mp/2} \simeq 2^{-Mp}$. Omschrijven geeft dat dit overeenkomt met een fout van $N^{-p/n}$. □

Onze implementatie

3.8 Tensorproduct

Herinner de gelijkheid

$$V_j = V_{j+1} \oplus W_{j+1}.$$

Dit kan ook herhaaldelijk toegepast worden:

$$V_j = V_{j+1} \oplus W_{j+1} = (V_{j+2} \oplus W_{j+2}) \oplus W_{j+1} = \ldots = V_L \oplus W_L \oplus \cdots \oplus W_{j+1}.$$

Dus we kunnen nu V_j^2 ook anders schrijven:

$$V_j^2 = V_j \otimes V_j = (V_L \oplus W_L \oplus \cdots \oplus W_{j+1}) \otimes (V_L \oplus W_L \oplus \cdots \oplus W_{j+1}).$$

Voor V_L was $\{\phi_{L,k} : k \in \mathbb{Z}\}$ een basis. Voor W_i was $\{\psi_{i,k} : k \in \mathbb{Z}\}$ een basis. Zo krijgen we dat

$$\Psi := \{\phi_{L,\cdot}, \psi_{L,\cdot}, \ldots, \psi_{j+1,\cdot}\}$$

een basis moet zijn voor V_j. Uit [4] T8.5 vinden we dat $\mathbf{\Psi} := \{\psi_1 \otimes \psi_2 : \psi_1, \psi_2 \in \Psi\}$ een basis is voor V_j^2. Dit kunnen we uitschrijven tot

$$\mathbf{\Psi} = \left\{ \begin{array}{cccc} \phi_{L,\cdot} \otimes \phi_{L,\cdot}, & \phi_{L,\cdot} \otimes \psi_{L,\cdot}, & \cdots, & \phi_{L,\cdot} \otimes \psi_{j+1,\cdot}, \\ \psi_{L,\cdot} \otimes \phi_{L,\cdot}, & \psi_{L,\cdot} \otimes \psi_{L,\cdot}, & \cdots, & \psi_{L,\cdot} \otimes \psi_{j+1,\cdot}, \\ \vdots & & & \\ \psi_{j+1,\cdot} \otimes \psi_{j+1,\cdot}, & \phi_{L,\cdot} \otimes \psi_{L,\cdot}, & \cdots, & \psi_{j+1,\cdot} \otimes \psi_{j+1,\cdot} \end{array} \right\}.$$

In onze definitie[3] is het Tensorproduct nu het schrijven van het (tweedimensionale) signaal in termen van deze basis $\mathbf{\Psi}$.

Meer dan twee dimensies

Het moge duidelijk zijn hoe het Tensorproduct schaalt naar meer dimensies. De $\mathbf{\Psi}$ moet hier gewoon niet langer twee basisfuncties maar n basisfuncties pakken.

Fout van het Tensorproduct

Volgens [3] L3.1.7 is

$$\mathbf{\Psi} = \Psi \otimes \cdots \otimes \Psi = \{\boldsymbol{\psi}_{\boldsymbol{\lambda}} := \psi_{\lambda_1} \otimes \cdots \otimes \psi_{\lambda_n} : \lambda_i \in \nabla\}$$

met $\boldsymbol{\lambda} := (\lambda_1, \ldots, \lambda_n) \in \mathbf{\nabla} := \nabla^n$ nu een orthogonale basis voor $L^2(\square)$.

Laat vervolgens $I_M := \{l \in \mathbb{N}_0^n : ||l||_1 \leq M\}$ en maak de *sparse grid index set* $\mathbf{\nabla}_M := \{(\boldsymbol{j}, \boldsymbol{n}) \in \mathbf{\nabla} : \boldsymbol{j} \in I_M\}$.

Lemma. *[3] P3.2.3 Voor $f \in H^p(\square)$ geldt dat de fout van de benadering op basis van de sparse grid index set $\mathbf{\nabla}_M$ hoogstens voldoet aan*

$$\|f - f_{\mathbf{\nabla}_M}\|_{L_2(\square)} \lesssim 2^{-pM} M^{(n-1)/2} \|f\|_{H^p(\square)}$$

[3] In de literatuur wordt de Mallatdecompositie ook vaak genoeg een Tensorproduct genoemd. Dit is natuurlijk geen verkeerde (hoogstens een verwarrende) naamgeving want de Mallatdecompositie *is* gewoon een Tensorproduct.

Het aantal elementen in deze verzameling $\boldsymbol{\nabla}_M$ nu, kunnen we vinden.

Lemma. *[3] L3.3.1 Het aantal elementen in $\boldsymbol{\nabla}_M$ is proportioneel met $2^M M^{n-1}$.*

Lemma. *Wanneer er voor twee functies f, g geldt dat $f(J) \lesssim J^{-p} \log_2(J)^\mu$ en $g(J) = \log_2(J)^\nu J =: N$, dan*

$$(f \circ g^{-1})(N) \lesssim N^{-p} \log_2 N^{\mu+\nu p}.$$

TODO: bewijs klopt (nog) niet

Met bovenstaande drie lemma's is het nu mogelijk een goede afschatting te maken.

Stelling (Fout van het Tensorproduct)**.** *Laat $f \in H^p(\square)$ en $N = \#\boldsymbol{\nabla}_M$. Dan:*

$$\|f - f_{\boldsymbol{\nabla}_M}\|_{L_2(\square)} \lesssim N^{-p} \log_2(N)^{(n-1)(1/2+p)} \|f\|_{H^p(\square)}.$$

Bewijs. Gebruik het tweede lemma om te vinden dat $N \simeq M^{n-1}2^M$. Nu vinden we via het eerste lemma dat

$$\|f - f_{\boldsymbol{\nabla}_M}\|_{L_2(\square)} \lesssim M^{(n-1)/2} 2^{-Mp} \|f\|_{H^p(\square)}$$

zodat

$$\frac{\|f - f_{\boldsymbol{\nabla}_M}\|_{L_2(\square)}}{\|f\|_{H^p(\square)}} \lesssim M^{(n-1)/2} 2^{-Mp}.$$

We willen graag lemma drie toepassen. Door

$$J^{-p} \log_2(J)^\mu = M^{(n-1)/2} 2^{-Mp}$$

volgt $J = 2^m$ en $\mu = (n-1)/2$. Door

$$N = M^{n-1}2^M = \log_2(J)^\nu J = 2^M M^\nu$$

volgt $\nu = n - 1$. TODO: wat is f en g? □

We vinden dus dat voor een voldoend gladde f dat (gebruikend een beperkte hoeveelheid basisfuncties) de Mallatdecompositie hoogstens een convergentiesnelheid $N^{-p/n}$ bereikt, terwijl het Tensorproduct theoretisch een snelheid $N^{-p} \log_2(N)^{(n-1)(p+1/2)}$ bereikt. Let op dat voor $n = 1$ de twee convergentiesnelheden beide N^{-p} zijn (omdat beide bases dan gewoon hetzelfde zijn).

In de praktijk hebben we echter nooit te maken met compleet gladde functies. Gevolg is dat deze stellingen niet helemaal opgaan. Niet *helemaal*, omdat per constructie van onze wavelet, de basis compact is en dus 'zoomen we in' op de functie. Lokaal is de mogelijkheid dat f glad genoeg is ineens een stuk meer in zicht. Het gevolg is wel dat je op zo'n moment waarschijnlijk lokaal een hoger niveau wil gebruiken. De bewijzen van hierboven zijn op basis van een *niet-adaptieve* deelverzameling, dat wil zeggen dat ze geen rekening houden met lokaal een hoger niveau.

Hoewel de vorige zin misschien klinkt alsof er roet in ons eten gegooid wordt, is het in de praktijk toch goed mogelijk om de gevolgen te zien. Dit zullen we zien wanneer we de twee decomposities zullen gaan vergelijken.

Onze implementatie

Hoofdstuk 4

Resultaten

4.1 Fouriertransformatie

Praktische nadelen

Geluid

Plaatjes

4.2 Wavelettransformaties d.m.v. Mallatdecompositie

Geluid

Plaatjes

4.3 Wavelettransformaties in 3D: film

Filmpjes met Mallat

Filmpjes met Tensor

Hoofdstuk 5

Reflectie en discussie

Hoofdstuk 6

Populaire samenvatting

Bibliografie

[1] Mohamad Akra, Louay Bazzi, *On the solution of linear recurrence equations*. Computational Optimization and Applications, 10(2):195 - 210, 1998.

[2] Bochner S., Chandrasekharan K. *Fourier Transforms*. Princeton University Press. 1949.

[3] Tammo Jan Dijkstra, *Adaptive tensor product wavelet methods for solving PDEs*, 2009

[4] http://www.uio.no/studier/emner/matnat/math/MAT-INF2360/v12/tensorwavelet.pdf

[5] Stéphane Mallat, *A Wavelet Tour of Signal Processing*

[6] http://www.ams.org/journals/bull/1960-66-02/S0002-9904-1960-10426-0/S0002-9904-1960-10426-0.pdf