

wavelet analysis (آنالیز موجک)

ایمان الیاسیان، دانشجوی دکترای عمران سازه eiman,elyasian@gmail.com

ایده ی نمایش یک تابع برحسب مجموعه ی کاملی از توابع اولین بار توسط ژوزف فوریه، ریاضیدان و فیزیکدان بین سال های ۱۸۰۶-۱۸۰۲ طی رساله ای در آکادمی علوم راجع به انتشار حرارت، برای نمایش توابع بکار گرفته شد. در واقع برای آنکه یک تابع $f(x)$ به شیوه ای ساده و فشرده نمایش داده شود فوریه اساساً ثابت کرد که می توان از محور هایی استفاده کرد که بکمک مجموعه ایی نامتناهی از توابع سینوس وار ساخته می شوند. عبارت دیگر فوریه نشان داد که یک تابع $f(x)$ را می توان بوسیله ی حاصل جمع بی نهایت تابع سینوسی و کسینوسی به شکل $\sin(ax)$ و $\cos(ax)$ نمایش داد. پایه های فوریه بصورت ابزار هایی اساسی، با کاربردهای فوق العاده متواتر در علوم، در آمده اند، زیرا برای نمایش انواع متعددی از توابع و در نتیجه کمین های فیزیکی فراوان بکار می روند. با گذشت زمان ضعف پایه های فوریه نمایان شد مثلاً دانشمندان پی بردند پایه های فوریه و نمایش توابع سینوس وار در مورد سیگنال های پیچیده نظری تصاویر، نه تنها ایده آل نیستند بلکه از شرایط مطلوب دورند، بعنوان مثال به شکل کارآمدی قادر به نمایش ساختارهای گذرا نظیر مرزهای موجود در تصاویر نیستند. همچنین آنها متوجه شدند تبدیل فوریه فقط برای توابع پایه مورد استفاده قرار می گیرد و برای توابع غیر پایه کار آمد نیست. (البته در سال ۱۹۴۶ با استفاده از توابع پنجره ای، که منجر به تبدیل فوریه ی پنجره ای شداین مشکل حل شد.) در سال ۱۹۰۹ هار اولین کسی بود که به موجک ها اشاره کرد. در سال های ۱۹۳۰ ریاضیدانان به قصد تحلیل ساختارهای تکین موضوعی به فکر اصلاح پایه های فوریه افتادند. و بعد از آن در سال ۱۹۷۰ یک ژئوفیزیکدان فرانسوی به نام ژان مورله متوجه شد که پایه های فوریه بهترین ابزار ممکن در اکتشافات زیر زمین نیستند، این موضوع در آزمایشگاهی متعلق به الف آکیلن منجر به یکی از اکتشافات تبدیل به موجک ها گردید. در سال ۱۹۸۰ ایومیر ریاضیدان فرانسوی، نخستین پایه های موجکی متعامد را کشف کرد (تعامد نوعی از ویژگی ها را بیان می کند که موجب تسهیلات فراوانی در استدلال و محاسبه می شود، پایه های فوریه نیز متعامدند.) در همین سال ها مورله مفهوم موجک و تبدیل موجک را بعنوان یک ابزار برای آنالیز سیگنال زمین لرزه وارد کرد و گراسمن فیزیکدان نظری فرانسه نیز فرمول وارونی را برای تبدیل موجک بدست آورد. در سال ۱۹۷۶ میرو و مالت از پایه های موجک متعامد توانستند آنالیز چند تفکیکی را بسازند و مالت تجزیه موجک ها و الگوریتم های بازسازی را با بکار بردن آنالیز چند تفکیکی بوجود آورد. در سال ۱۹۹۰ مورنزی همراه با آنتوان موجک ها را به دو بعد و سپس به فضاهایی با ابعاد دیگر گسترش دادند و بدین ترتیب بود که آنالیز موجکی پایه گذاری گردید.

آشنایی با تحلیل موجک

آنالیز موجک (Wavelet Analysis) یکی از دستاوردهای نسبتاً جدید و هیجان انگیز ریاضیات محض که مبتنی بر چندین دهه پژوهش در آنالیز همساز است، امروزه کاربردهای مهمی در بسیاری از رشته های علوم و مهندسی یافته و امکانات جدیدی برای درک جنبه های ریاضی آن و نیز افزایش کاربردهایش فراهم شده است. در آنالیز موجک هم مانند آنالیز فوریه با بسط تابع ها سروکار داریم ولی این بسط برحسب «موجک ها»

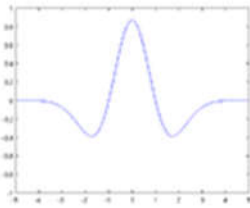
انجام می شود. موجک تابع مشخص مفروضی با میانگین صفر است و بسط برحسب انتقالها و اتساعهای این تابع انجام می گیرد، بر خلاف چند جمله ای های مثلثاتی، موجک ها در فضا بصورت موضعی بررسی می شوند و به این ترتیب ارتباط نزدیکی بین بعضی توابع و ضرایب آن ها امکان پذیر می شود و پایداری عددی بیشتری در باز سازی و محاسبات فراهم می گردد. هر کاربردی را که مبتنی بر تبدیل سریع فوریه است می توان با استفاده از موجک ها فومول بندی کرد و اطلاعات فضایی (یا زمانی) موضعی بیشتری بدست آورد. بطور کلی، این موضوع بر پردازش سیگنال و تصویر و الگوریتم های عددی سریع برای محاسبه ی عملگرهای انتگرالی اثر می گذارد. آنالیز موجک حاصل ۵۰ سال کار ریاضی (نظریه ی لیتلود – پیلی و کالدرون – زیگموند) است که طی آن، با توجه به مشکلاتی که در پاسخ دادن به ساده ترین پرسش های مربوط به تبدیل فوریه وجود داشت، جانشینهای انعطاف پذیر ساده تری از طریق آنالیز همساز ارائه شدند. مستقل از این نظریه که درون ریاضیات محض جای دارد، صورتهای مختلفی از این رهیافت چند مقیاسی (multi Scale) را در طی دهه ی گذشته در پردازش تصویر، آکوستیک، کدگذاری (به شکل فیلترهای آینه ای متعامد و الگوریتمهای هرمی)، و استخراج نفت دیده ایم.

کاربردهای تحلیل موجک

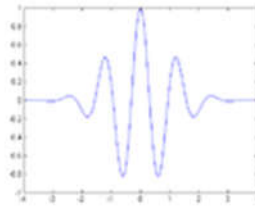
آنالیز موجک همراه با تبدیل سریع فوریه در تحلیل سیگنالهای گذرای که سریعاً تغییر می کنند، صدا و سیگنالهای صوتی، جریان های الکتریکی در مغز، صداهای زیر آبی ضربه ای و داده های طیف نمایی NMR، و در کنترل نیروگاههای برق از طریق صفحه ی نمایش کامپیوتر بکار رفته است. و نیز بعنوان ابزاری علمی، برای روشن ساختن ساختارهای پیچیده ای که در تلاطم ظاهر می شوند، جریان های جوی، و در بررسی ساختارهای ستاره ای از آن استفاده شده است. این آنالیز به عنوان یک ابزار عددی می تواند مانند تبدیل سریع فوریه تا حد زیادی از پیچیدگی محاسبات بزرگ مقیاس بکاهد، بدین ترتیب که با تغییر هموار ضریب، ماتریس های متراکم را به شکل تنکی که به سرعت قابل محاسبه باشد در آورد. راحتی و سادگی این آنالیز باعث ساختن تراشه هایی شده است که قادر به کدگذاری به نحوی بسیار کارا، و فشرده سازی سیگنالها و تصاویرند. آنالیز موجک امروزه کاربردهای فراوانی پیدا کرده است که از آن جمله می توان به کاربرد آن در تصویر برداری پزشکی (MRI) و سی تی اسکن (CAT)، جداسازی بافت های مغزی از تصاویر تشدید مغناطیسی، تشخیص خودکار خوشه های میکروکلسیفیکاسیون، تحلیل تصاویر طیفی تشدید مغناطیسی (MR Spectroscopy) و عملکردهای تشدید مغناطیسی (F MRI) اشاره کرد.

موجک

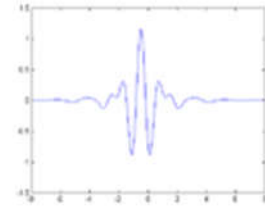
موجک (Wavelet) دسته ای از توابع ریاضی هستند که برای تجزیه سیگنال پیوسته به مؤلفه های فرکانسی آن بکار می رود که رزولوشن هر مؤلفه برابر با مقیاس آن است. تبدیل موجک تجزیه یک تابع بر مبنای توابع موجک می باشد. موجک ها (که به عنوان موجک های دختر شناخته می شوند) نمونه های انتقال یافته و مقیاس شده یک تابع (موجک مادر) با طول متناهی و نوسانی شدیداً میرا هستند. چند نمونه موجک مادر در شکل زیر نمایش داده شده اند.



کلاه مکزیکی



مورله



میر

تبدیل‌های موجک

تعداد زیادی تبدیل موجک وجود دارد که لیست آن را می‌شود در فهرست تبدیل‌های مرتبط با موجک مشاهده نمود. معمول‌ترین این تبدیل‌ها عبارتند از:

- تبدیل موجک پیوسته (Continuous wavelet transform (CWT)
- تبدیل موجک گسسته (Discrete wavelet transform (DWT)
- تبدیل سریع موجک (Fast wavelet transform (FWT)
- Lifting scheme
- تجزیه بسته‌های موجک (Wavelet packet decomposition (WPD)
- تبدیل موجک ساکن (Stationary wavelet transform (SWT)

موجک‌ها و معادلات اتساع

موجک‌ها بر مبنای دو عمل اصلی قرار دارند:

- انتقال (Translation)

$$W(x) \rightarrow W(x + a)$$
- اتساع (Dilation)

$$W(x) \rightarrow W(bx)$$

مقایسه با تبدیل فوریه

در مقایسه با تبدیل فوریه می‌توان گفت که تبدیل موجک دارای خصوصیت محلی‌سازی بسیار خوبی است. بطور مثال تبدیل فوریه یک پیک تیز دارای تعداد زیادی ضریب است، چرا که توابع پایه تبدیل فوریه توابع سینوسی و کسینوسی هستند که دامنه آنها در کل بازه ثابت است، در حالی که توابع موجک توابعی هستند که بیشتر انرژی آنها در بازه کوچکی متمرکز شده است و به سرعت میرا می‌شوند. بنابراین با انتخاب مناسب موجک‌های مادر می‌توان فشردگی بیشتری در مقایسه با تبدیل فوریه انجام داد.

تاریخچه

در تاریخ ریاضیات مبادی و ریشه‌های متعددی را می‌توان برای موجک‌ها سراغ گرفت.

کارهای قبل از ۱۹۳۰

مربوط به قبل از ۱۹۳۰ (م) می‌توان به آنالیز فرکانس‌ها اشاره کرد، که به وسیلهٔ فوریه شروع شد.

استفاده از واژهٔ موجک‌ها، برای اولین بار، در یکی از ضمیمه‌های تز آلفرد هار (۱۹۰۹ م) ظاهر شد. امروزه هم، این موجک‌ها به همان نام یعنی به موجک‌های هار معروف اند. موجک‌های هار دارای دامنهٔ تعریف فشرده (compact) بوده، و غیر مشتق‌پذیر به صورت پیوسته هستند.

کارهای مربوط به دهه ۱۹۳۰

در این دهه چند گروه پیرامون موضوع نمایش توابع با به کارگیری پایه‌های با مقیاس متغیر برای تنیدن فضاهای توابع تحقیق می‌نمودند.

موجک‌های متعامد

با دیدی کلی می‌توان اظهار داشت که پایه‌های متعامد حالتی بهینه برای تنیدن فضاهای برداری (چه فضاهای با ابعاد متناهی و چه فضاهای بی نهایت بعدی) و انجام محاسبات ارائه می‌نمایند. لذا همواره تمایل و تلاش در این راستا قرار داشته که یا مجموعه پایه‌ها از آغاز متعامد انتخاب شود و یا آن که با شیوه‌هایی نظیر گرام اشمیت آنها را به سوی تعامد سوق داد.

موجک هار

موجک هار اولین موجک شناخته شده می‌باشد که پیدایش آن به سالهای ابتدای قرن بیستم باز می‌گردد. این موجک ساده‌ترین نوع هم هست و پایه‌هایی متعامد برای تنیدن فضای محاسبه را ارائه می‌دهد.

شبکه عصبی ویولت چیست ؟

ویولت یا موجک یکی از ابزارهای مهم در مهندسی می باشد که در سایت گروه برنامه نویسی متلب ایران می توانید مطالب مفیدی را در این زمینه بیابید.

مطلبی که می خواهیم در این صفحه در مورد آن صحبت کنیم ترکیب ویولت و شبکه عصبی می باشد. ابتدا توضیح مختصری در مورد شبکه عصبی می آوریم.

شبکه عصبی چند لایه MLP

شبکه (MLP) مجموعه ای از نورون ها است که در لایه مختلفی پشت سر هم قرار گرفته اند. مقادیر ورودی پس از ضرب در وزن های موجود در گذر گاه های بین لایه ها به نورون بعدی رسیده و در آن جا با هم جمع می شوند و پس از عبور از تابع شبکه مربوطه خروجی نورون ها را تشکیل می دهند. در پایان خروجی به دست آمده با خروجی مورد نظر مقایسه شده و خطای به دست آمده جهت اصلاح وزن های شبکه به کار می رود ، این امر اصطلاحاً آموزش شبکه عصبی نامیده می شود .

قاعده فراگیری MLP

قاعده فراگیری پرسپترون چند لایه را «قاعده کلی دلتا» یا « قاعده پس انتشار » می گویند . این عناوین در سال 1986 رومل هارت ، مک کلند و ویلیامز پیشنهاد شد.

اولین گروهی بودند که نه تنها قاعده فراگیری پرسپترون را به طور مستقل کشف کردند بلکه با ترکیب آن ها پرسپترون چند لایه را ایجاد نمودند. کتاب آن ها به نام «پردازش توزیع شده موازی» یکی از مهم ترین منابع

این حوزه علمی می باشد. نحوه عمل پرسپترون چند لایه ای مشابه پرسپترون تک لایه ای است. به این صورت که الگویی به شبکه عرضه می شود و خروجی آن محاسبه می گردد ، مقایسه خروجی واقعی و خروجی مطلوب باعث می گردد که ضرایب وزنی شبکه تغییر یابد به طوری که در مراحل بعد خروجی صحیح تری حاصل شود. وقتی به شبکه آموزش ندیده ای الگویی را عرضه می کنیم ، خروجی های تصادفی تولید می کند. ابتدا باید تابع خطایی را تعریف کنیم که تفاوت خروجی واقعی و خروجی مطلوب را نشان دهد. از آن جایی که خروجی مطلوب را می دانیم این نوع فراگیری را « فراگیری با سرپرستی » می نامیم.

برای موفقیت در آموزش شبکه باید خروجی آن را به تدریج به خروجی مطلوب نزدیک کنیم. به عبارت دیگر باید میزان خطا را کاهش دهیم. برای رسیدن به هدف از قانون دلتا استفاده می کنیم.

تبدیل موجک (ویولت Wavelet)

تبدیلات موجک یا ویولت (Wavelet) از جمله ابزارهایی هستند که کاربردهای فراوانی در شاخه های مختلف علمی و مهندسی، به ویژه هوش مصنوعی، یادگیری ماشینی، پیش بینی سری زمانی، و بازشناسی الگو دارد. تئوری موجک ها در واقع تعمیمی بر تئوری تبدیلات و سری های فوریه است و ضعف های آنالیز فوریه در عملکرد موضعی و مدل سازی رفتارهای کوتاه مدت را، جبران می نماید.

موجک یا ویولت (به انگلیسی: Wavelet) دسته ای از توابع ریاضی هستند که برای تجزیه سیگنال پیوسته به مؤلفه های فرکانسی آن بکار می رود که رزولوشن هر مؤلفه برابر با مقیاس آن است. تبدیل موجک تجزیه یک تابع بر مبنای توابع موجک می باشد. موجک ها (که به عنوان موجک های دختر شناخته می شوند) نمونه های انتقال یافته و مقیاس شده یک تابع (موجک مادر) با طول متناهی و نوسانی شدیداً میرا هستند. چند نمونه موجک مادر در شکل زیر نمایش داده شده اند.

تعداد زیادی تبدیل موجک وجود دارد که لیست آن را می شود در فهرست تبدیل های مرتبط با موجک مشاهده نمود. معمول ترین این تبدیل ها عبارتند از:

- تبدیل موجک پیوسته (Continuous wavelet transform (CWT)
- تبدیل موجک گسسته (Discrete wavelet transform (DWT)
- تبدیل سریع موجک (Fast wavelet transform (FWT)
- Lifting scheme
- تجزیه بسته های موجک (Wavelet packet decomposition (WPD)
- تبدیل موجک ساکن (Stationary wavelet transform (SWT)

مقایسه تبدیل موجک با تبدیل فوریه

در مقایسه با تبدیل فوریه می توان گفت که تبدیل موجک دارای خصوصیت محلی سازی بسیار خوبی است. بطور مثال

تبدیل موجک (Wavelet Transform) یکی از تبدیلات مهم ریاضی است که در حوزه های مختلف علوم کاربرد دارد. ایده اصلی تبدیل موجک این است که بر ضعف ها و محدودیت های موجود در تبدیل فوریه غلبه

کند. این تبدیل را بر خلاف تبدیل فوری، می‌توان در مورد سیگنال‌های غیر ایستا و سیستم‌های دینامیک نیز مورد استفاده قرار داد. در این مطلب قصد داریم تا به بررسی تبدیل موجک و کاربرد آن در پردازش سیگنال و یادگیری ماشین بپردازیم. ابتدا مفاهیم، تئوری و عملکرد تبدیل موجک پیوسته و گسسته و تمایز آن با تبدیل فوری را بیان می‌کنیم و سپس مثال‌هایی کاربردی از اعمال تبدیل موجک روی دو دیتاست UCI-HAR و ECG و دسته‌بندی داده‌های آن‌ها با استفاده از الگوریتم‌های یادگیری ماشین و یادگیری عمیق در محیط پایتون را مورد بررسی قرار می‌دهیم. در نهایت، کدهای پیاده‌سازی تبدیل موجک در محیط متلب بدون استفاده از توابع آماده را بیان می‌کنیم و آن را روی دیتاست ال نینو پیاده‌سازی می‌کنیم.

1. از تبدیل فوری به تبدیل موجک

2. نحوه عملکرد تبدیل موجک

3. انواع مختلف تبدیل موجک

4. تبدیل موجک پیوسته و تبدیل موجک گسسته

4.1. تبدیل فوری گسسته به صورت بانک فیلتری

5. مصورسازی فضای حالت با استفاده از تبدیل موجک پیوسته

6. استفاده از تبدیل موجک در طبقه‌بندی سیگنال

6.1. بارگذاری دیتاست

6.2. اعمال تبدیل موجک پیوسته بر روی دیتاست

6.3. آموزش شبکه عصبی کانولوشنی

7. تجزیه سیگنال با استفاده از تبدیل موجک گسسته

8. حذف نویز (مولفه‌های فرکانس بالا) با استفاده از تبدیل موجک گسسته

9. استفاده از تبدیل موجک گسسته برای طبقه‌بندی سیگنال

9.1. ایده پنهان در طبقه‌بندی سیگنال با استفاده از تبدیل موجک گسسته

9.2. تولید ویژگی از هر زیر باند

9.3. استفاده از ویژگی‌ها و کتابخانه Scikit-Learn برای طبقه‌بندی دو دیتاست ECG

9.4. مقایسه دقت طبقه‌بندی با استفاده از تبدیل موجک گسسته، تبدیل فوری و شبکه عصبی بازگشتی

(RNN)

10. تبدیل موجک در متلب

10.1. wavetest.m

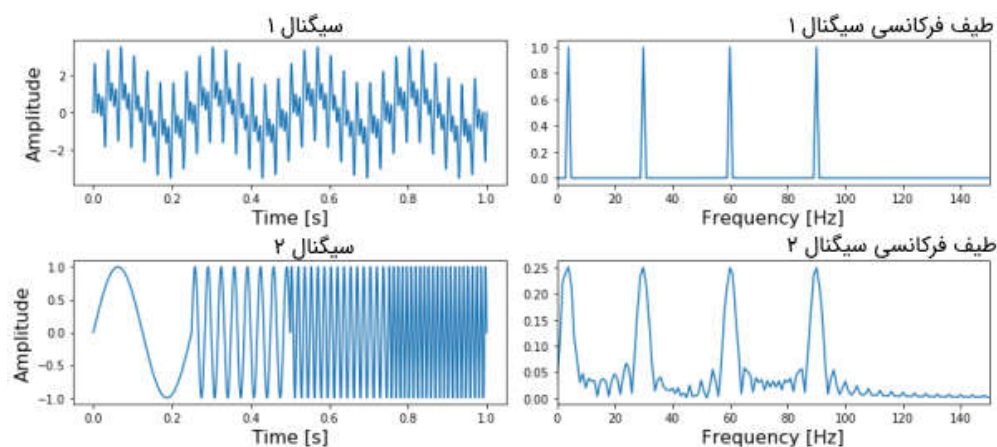
تبدیلات ریاضی کاربردهای فراوانی در پردازش و طبقه‌بندی داده‌های مختلف مانند سیگنال‌ها و سری‌های زمانی (Time Series) دارند. به عنوان مثال، با استفاده از تبدیل فوری (Fourier Transform) که در مطالب قبلی مجله فرادرس مورد بررسی قرار گرفته است، می‌توان یک سیگنال را از حوزه زمان به حوزه فرکانس منتقل کرد. پیک‌های ظاهر شده در نمودار طیف فرکانسی یک سیگنال پس از اعمال تبدیل فوری، نشان دهنده

فرکانس‌هایی است که در آن سیگنال غالب هستند. هر چقدر این پیک‌ها بزرگتر و تیزتر باشند، آن فرکانس در سیگنال حضور بیشتر و موثرتری دارد. محل قرار گرفتن پیک (مقدار فرکانس) و ارتفاع آن (دامنه فرکانس) در یک نمودار طیف فرکانسی، می‌توانند به عنوان ورودی برای الگوریتم‌هایی طبقه‌بندی مانند جنگل تصادفی (Random Forest) و یا ارتقای گرادیان (Gradient Boosting) مورد استفاده قرار گیرند. این تکنیک ساده برای بسیاری از مسائل عملکرد فوق‌العاده و دقت بسیار بالایی به همراه خواهد داشت. اما قاعده کلی که در مورد تبدیل فوریه وجود دارد این است که تبدیل فوریه تا زمانی که طیف فرکانسی یک سیگنال از لحاظ آماری ایستا (Stationary) باشد، به خوبی عمل خواهد کرد. ایستا بودن طیف فرکانسی به این معنی است که فرکانس‌های ظاهر شده در یک سیگنال، وابسته به زمان نباشند. به عبارت دیگر، اگر یک سیگنال شامل فرکانس X هرتر باشد، این فرکانس باید به صورت برابر در تمام طول سیگنال وجود داشته باشد. هرچه یک سیگنال بیشتر غیر ایستا یا دینامیک باشد، نتیجه بدتر خواهد بود. پردازش سیگنال‌های غیر ایستا معمولاً از دشواری بیشتری برخوردار است و برای آنالیز آن‌ها باید از تکنیک‌ها و پیش پردازش‌های دیگری استفاده کنیم. بسیاری از سیگنال‌های واقعی موجود در طبیعت دارای طبیعت غیر ایستا هستند. هنگامی که تلاش داریم سیگنال‌های ECG، بازار بورس، داده‌های تجهیزات و سنسورها و... را در پروژه‌های عملی مورد پردازش قرار دهیم، به احتمال زیاد با سیگنال‌های غیر ایستا و سیستم‌های دینامیک رو به رو خواهیم شد. یک راه حل بسیار عالی برای پردازش سیگنال‌های غیر ایستا، استفاده از تبدیل موجک به جای تبدیل فوریه است. با این که تبدیل موجک یک ابزار بسیار قدرتمند برای پردازش و طبقه‌بندی سیگنال‌ها و سری‌های زمانی دینامیک محسوب می‌شود، اما هنوز در علم داده به صورت گسترده مورد استفاده قرار نمی‌گیرد و از محبوبیت بالایی برخوردار نیست. احتمالاً دلیل این امر، نیاز به داشتن دانش اولیه درباره ریاضیات، تبدیل فوریه و پردازش سیگنال است، تا پس از آن، ریاضیات نهفته در تبدیل موجک به صورت کامل قابل درک باشد. البته بسیاری از مراجع و کتاب‌ها به صورت تئوری تبدیل موجک را مورد بررسی قرار داده‌اند، در حالی که دانش عملی کافی درباره نحوه استفاده از آن به ندرت مورد بحث قرار گرفته است. به همین دلیل، در این مطلب می‌خواهیم به بیان مباحث تئوری مرتبط با تبدیل موجک بپردازیم و با کاربردهای عملی تبدیل موجک آشنا شویم، اما به صورت عمیق ریاضیات آن را مورد بررسی قرار نمی‌دهیم. در هر گام از مطلب، کدهای پایتون را ارائه می‌دهیم، به صورتی که در پایان قادر باشید از تبدیل موجک در پروژه‌های خود استفاده کنید. در کدهایی که در ادامه نوشته می‌شود، از پکیج PyWavelets استفاده شده است، بنابراین توصیه می‌شود تا این پکیج را ابتدا با دستور «pip install pywavelets» نصب نمایید.

از تبدیل فوریه به تبدیل موجک

همان طور که می‌دانیم، تبدیل فوریه از طریق ضرب کردن سیگنال مورد پردازش در قطاری از سیگنال‌های سینوسی با فرکانس‌های مختلف عمل می‌کند. در واقع، از این راه می‌توانیم تعیین کنیم که کدام فرکانس‌ها در سیگنال مورد پردازش وجود دارند. اگر عملگر ضرب نقطه‌ای بین سیگنال مورد نظر و یک سیگنال سینوسی با فرکانس مشخص، برابر با یک عدد با دامنه بزرگ شود، آن‌گاه می‌توان نتیجه گرفت که هم‌پوشانی زیادی بین

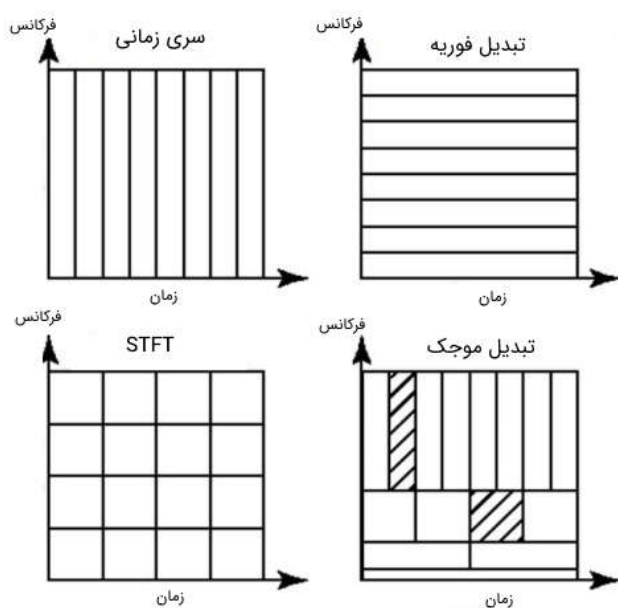
این دو سیگنال وجود دارد و در نتیجه آن فرکانس مشخص در طیف فرکانسی سیگنال مورد نظر نیز مشاهده خواهد شد. قطعا دلیل این امر از آنجایی ناشی می‌شود که عملگر ضرب نقطه‌ای معیاری برای اندازه‌گیری میزان هم‌پوشانی و شباهت بین دو بردار یا دو سیگنال است. نکته‌ای که در مورد تبدیل فوریه می‌توان به آن اشاره کرد این است که در حوزه فرکانس دارای رزولوشن بالایی است، در حالی که در حوزه زمان از رزولوشن صفر برخوردار است. به عبارت دیگر، تبدیل فوریه این توانایی را دارد که به ما بگوید دقیقا چه فرکانس‌هایی در یک سیگنال وجود دارند، اما نمی‌توان با استفاده از آن تعیین کرد که فرکانس مورد نظر در چه لحظه‌ای از زمان در سیگنال اتفاق می‌افتد. این مفهوم را می‌توان به سادگی با استفاده از قطعه کد زیر به نمایش در آورد: خروجی این قطعه کد در تصویر زیر دیده می‌شود. در این تصویر سیگنال‌های اصلی به همراه طیف فرکانسی هر کدام نشان داده شده است.



سیگنال‌های اصلی به همراه طیف فرکانسی هر یک

در تصاویر بالا، در ردیف اول سیگنال اصلی به همراه طیف فرکانسی آن دیده می‌شود که شامل چهار فرکانس در تمام زمان‌ها است و در ردیف پایین می‌توان دید که سیگنال دوم نیز چهار فرکانس مختلف اما در چهار زمان متفاوت دارد. در این تصویر، اولین سیگنال از سمت چپ، بر اساس تصویر طیف فرکانسی خود، چهار فرکانس مختلف در ۶۰، ۳۰، ۴ و ۹۰ هرتز دارد. این فرکانس‌ها در تمام زمان‌ها وجود دارند. در سیگنال دوم نیز چهار فرکانس مشابه وجود دارد، اما تفاوتی که دارد این است که فرکانس اول فقط در ربع اول از سیگنال، فرکانس دوم در ربع دوم سیگنال، فرکانس سوم در ربع سوم و فرکانس چهارم فقط در ربع آخر سیگنال وجود دارند. نکته مهمی که در اینجا باید به آن توجه شود این است که هر دو نمودار طیف فرکانسی در تصویر بالا، دارای چهار پیک دقیقاً یکسان در ۴ فرکانس ذکر شده هستند. بنابراین با استفاده از تبدیل فوریه نمی‌توان به این واقعیت پی برد که فرکانس در سیگنال اصلی دقیقاً در کجا اتفاق می‌افتد. دقیقاً به همین دلیل است که در مثال بالا نیز تبدیل فوریه نتوانست به ما در تمایز و تشخیص دو سیگنال از یکدیگر کمک کند. توجه کنید که وقوع لبه‌های گذرا جانبی در طیف فرکانسی سیگنال دوم، به دلیل ناپیوستگی بین چهار فرکانس در سیگنال است. برای غلبه بر این مشکل، روش تبدیل فوریه زمان کوتاه (Short-Time Fourier Transform) یا STFT مورد استفاده قرار می‌گیرد. در این روش، سیگنال اصلی به چندین بخش با طول یکسان تقسیم می‌شوند.

این بخش‌ها ممکن است با یکدیگر هم‌پوشانی داشته باشند و یا فاقد هم‌پوشانی باشند. با استفاده از پنجره‌های لغزشی (Sliding Window)، سیگنال را قبل از اعمال تبدیل فوریه به چندین بخش تقسیم می‌کنیم. ایده اصلی در این روش بسیار ساده است. اگر سیگنال را مثلاً به 10 قسمت تقسیم کرده باشیم و تبدیل فوریه در بخش دوم فرکانس خاصی را تشخیص دهد، بنابراین با قطعیت بالا می‌توان گفت که این فرکانس در بازه بین 210210 و 310310 از سیگنال اصلی به وقوع می‌پیوندد. اما عیب اصلی که در این روش وجود دارد این است که با یک محدودیت فیزیکی در تبدیل فوریه رو به رو خواهد شد که عدم قطعیت نام دارد. در این روش، هرچه اندازه پنجره‌ها را کوچک‌تر کنیم، قادر خواهیم بود به صورت دقیق‌تر تعیین کنیم که یک فرکانس در چه زمانی از سیگنال اصلی به وقوع پیوسته است، اما از طرف دیگر اطلاعات کم‌تری را راجع به مقدار فرکانس سیگنال اصلی به دست خواهیم آورد. به صورت مشابه، هر چه اندازه پنجره‌ها را بزرگ‌تر انتخاب کنیم، اطلاعات بیشتری راجع به مقدار فرکانس و اطلاعات کم‌تری راجع به زمان وقوع فرکانس به دست خواهیم آورد. روش بهتری که برای آنالیز یک سیگنال با طیف فرکانسی دینامیک وجود دارد، استفاده از تبدیل موجک است. تبدیل موجک هم در حوزه زمان و هم در حوزه فرکانس دارای رزولوشن بالایی است. این تبدیل نه تنها مقدار فرکانس‌های موجود در سیگنال را مشخص می‌کند، بلکه تعیین می‌کند که آن فرکانس‌ها در چه زمانی از سیگنال به وقوع می‌پیوندند. تبدیل موجک این توانایی را از طریق کار کردن در مقیاس‌های (Scale) مختلف به دست می‌آورد. در تبدیل موجک، ابتدا سیگنال را با مقیاس یا پنجره بزرگ در نظر می‌گیریم و ویژگی‌های بزرگ (Large Features) آن را آنالیز می‌کنیم. در گام بعد، با پنجره‌های کوچک به سیگنال نگاه می‌کنیم و ویژگی‌های کوچک سیگنال را به دست می‌آوریم. در تصویر زیر رزولوشن حوزه زمان و فرکانس در روش تبدیل‌های مختلف به نمایش درآمده است.



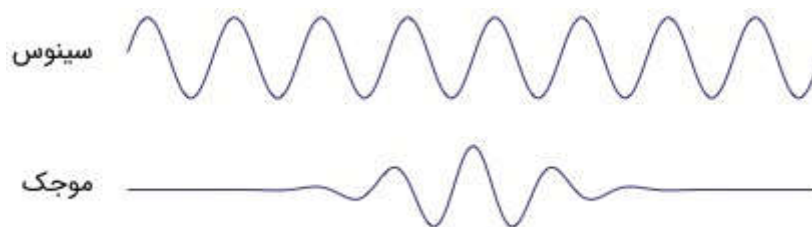
در تصویر بالا، اندازه و جهت بلوک‌ها نشان‌دهنده مقدار رزولوشن در آن تبدیل است، به عبارت دیگر بلوک‌ها در هر تبدیل تعیین می‌کنند که در حوزه زمان و فرکانس می‌توان ویژگی‌های تا چه مقدار کوچک را با استفاده از آن تبدیل تشخیص داد. سیگنال اصلی که در حوزه زمان است، دارای رزولوشن بالا در حوزه زمان و رزولوشن صفر در حوزه فرکانس است. این ویژگی به این معنی است که می‌توانیم ویژگی‌های بسیار کوچکی را در حوزه زمان تشخیص دهیم، اما در حوزه فرکانس هیچ ویژگی قابل تمایز نیست.

اما تبدیل فوریه زمان کوتاه، دارای رزولوشن با اندازه متوسط در هر دو حوزه زمان و فرکانس است. رزولوشن تبدیل موجک به صورت زیر تغییر می‌کند:

برای مقادیر فرکانس‌های کوچک، رزولوشن بالا در حوزه فرکانس و رزولوشن پایین در حوزه زمان دارد. برای مقادیر فرکانس‌های بالا، رزولوشن پایین در حوزه فرکانس و رزولوشن بالا در حوزه زمان دارد. بنابراین در حالت کلی می‌توان گفت تبدیل موجک به صورت مصالحه‌ای عمل می‌کند. در مقیاس‌هایی که مشخصه‌های وابسته به زمان جذاب‌تر هستند، تبدیل موجک دارای رزولوشن بالاتر در حوزه زمان و در مقیاس‌هایی که مشخصه‌های وابسته به فرکانس جذاب‌تر هستند، دارای رزولوشن بالاتر در حوزه فرکانس است. این نوع مصالحه دقیقاً همان هدفی است که در پردازش سیگنال مورد نظر است.

نحوه عملکرد تبدیل موجک

همان طور که اشاره کردیم، تبدیل فوریه برای آنالیز سیگنال از یک سری امواج سینوسی با فرکانس‌های مختلف استفاده می‌کند. در این حالت، سیگنال به صورت ترکیبی خطی از سیگنال‌های سینوسی نمایش داده می‌شود. اما تبدیل موجک از تعدادی توابع به نام موجک استفاده می‌کند که هر کدام مقیاس متفاوتی دارند. همان طور که می‌دانیم معنی واژه موجک، موج کوچک است و توابع موجک نیز دقیقاً به همین صورت کوچک هستند. در تصویر زیر تفاوت بین یک سیگنال سینوسی و یک موجک نشان داده شده است.



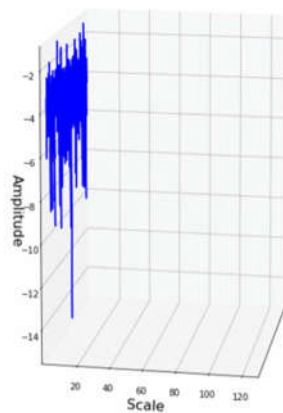
با دقت در تصویر بالا، کاملاً مشخص است که سیگنال سینوسی در یک لحظه خاص از زمان واقع نشده است. این سیگنال از بی‌نهایت شروع می‌شود و تا بی‌نهایت ادامه می‌یابد، در حالی که یک موجک در لحظه خاصی از زمان واقع شده است. این ویژگی به تبدیل موجک اجازه می‌دهد تا علاوه بر اطلاعات فرکانسی، اطلاعات زمانی را نیز به دست آورد.

چون موجک در زمان واقع شده است، در نتیجه می‌توان سیگنال اصلی را در لحظات مختلف از زمان در موجک ضرب کرد. در گام نخست، با نقاط ابتدایی سیگنال شروع می‌کنیم و به تدریج موجک را به سمت انتهای سیگنال حرکت می‌دهیم. این عمل را کانولوشن (Convolution) می‌گویند. بعد از این که کانولوشن را با

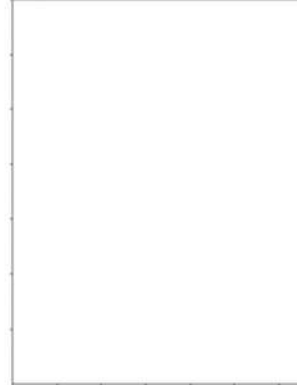
سیگنال موجک اصلی (موجک مادر) انجام دادیم، می‌توانیم آن را به نحوی مقیاس‌دهی کنیم که بزرگ‌تر شود و دوباره فرایند را تکرار کنیم. این فرایند در تصویر متحرک زیر نشان داده شده است.

$$\text{Signal (S)} * \text{Scaled Wavelet (SW)} = \text{Convolution(S, SW)}$$


3D plot of Wavelet Transform



2D plot of Wavelet Transform



source: ataspinar.com

همان طور که از تصویر بالا مشخص است، تبدیل موجک یک سیگنال تک بعدی، دارای دو بعد است. این خروجی دو بعدی مربوط به تبدیل موجک، نمایش سیگنال اصلی بر حسب مقیاس و زمان است که به طیف اسپکتروگرام (Spectrogram) یا اسکالوگرام (Scaleogram) معروف است. در تصویر بالا، ابتدا تبدیل موجک به صورت سه بعدی و سپس به صورت دو بعدی نیز ترسیم شده است. ممکن است این سوال پیش بیاید که بُعد مقیاس (Scale) در تصویر اسپکتروگرام نشان‌دهنده چیست؟ در واقع، چون کلمه فرکانس اغلب برای تبدیل فوریه مورد استفاده قرار می‌گیرد، تبدیل موجک اغلب با مقیاس بیان می‌شود. در نتیجه در تبدیل موجک، دو بُعد اسپکتروگرام، زمان و مقیاس هستند. در مسائلی که فرکانس مناسب‌تر و شهودی‌تر از مقیاس باشد، می‌توان از طریق فرمول زیر، مقیاس را به شبه‌فرکانس (Pseudo-Frequencies) تبدیل کرد:

در این فرمول، f_{afa} برابر با شبه‌فرکانس، f_{cfc} فرکانس مرکزی سیگنال موجک مادر و a_a فاکتور مقیاس (Scaling Factor) است. واضح است که فاکتور مقیاس بالاتر (موجک طولانی‌تر) متناظر با فرکانس‌های پایین‌تر است، بنابراین با مقیاس‌دهی به سیگنال موجک در حوزه زمان، می‌توانیم فرکانس‌های کوچک‌تر را آنالیز کنیم و در نتیجه به رزولوشن بالاتر در حوزه فرکانس می‌رسیم. به طریق مشابه، با استفاده از مقیاس کوچک‌تر، رزولوشن در حوزه زمان بالاتر خواهد رفت. بنابراین می‌توان گفت مقیاس، معکوس فرکانس در تبدیل فوریه است. پکیج PyWavelets در پایتون، شامل تابع `scale2frequency` است تا تبدیل را از حوزه مقیاس به حوزه فرکانس انجام دهد.

انواع مختلف تبدیل موجک

تفاوت دیگری که بین تبدیل فوریه و تبدیل موجک وجود دارد، این است که تبدیل موجک انواع مختلفی دارد. برای انواع مختلف تبدیل موجک، مصالحه بین فشردگی (Compact) و صاف بودن (Smooth) با یکدیگر تفاوت دارند. این ویژگی بدین معنی است که می‌توانیم نوع خاصی از تبدیل موجک را انتخاب کنیم که با ویژگی (Features) مورد نظر برای استخراج از سیگنال تناسب بیشتری داشته باشد. به عنوان مثال کتابخانه PyWavelets شامل ۱۴ موجک مادر (انواع موجک) است. در قطعه کد زیر انواع این خانواده موجک‌ها را می‌توان مشاهده کرد. هر خانواده از موجک‌ها دارای شکل، فشردگی و همواری متفاوتی هستند و برای هدف متمایزی مورد استفاده قرار می‌گیرد. چون فقط دو شرط ریاضی وجود دارد که تابع موجک باید در آن صدق کند، در نتیجه تولید انواع مختلف موجک کار آسانی است. دو شرط ریاضی که موجک باید در آن‌ها صدق کند، قیود متعامدسازی (Orthogonalization) و نرمال‌سازی (Normalization) نام دارند. بر اساس این قیود، یک موجک باید اولاً انرژی محدود (Finite Energy) داشته باشد و دوماً میانگین آن برابر با صفر باشد. انرژی محدود به این معنی است که سیگنال در زمان و فرکانس متمرکز (Localized) باشد، انتگرال‌پذیر باشد و ضرب داخلی بین سیگنال و موجک همیشه وجود داشته باشد. همچنین بر اساس شرط مقبولیت دوم، موجک باید دارای میانگین صفر در حوزه زمان باشد یا به عبارت دیگر، در حوزه زمان و در فرکانس صفر، مقدار آن برابر با صفر باشد. این شرط برای اطمینان از انتگرال‌پذیری و نیز محاسبه معکوس تبدیل موجک بسیار ضروری است. علاوه بر این داریم:

یک موجک می‌تواند متعامد (Orthogonal) یا غیر متعامد باشد.

یک موجک می‌تواند متعامد دو طرفه (Bi-Orthogonal) باشد یا نباشد.

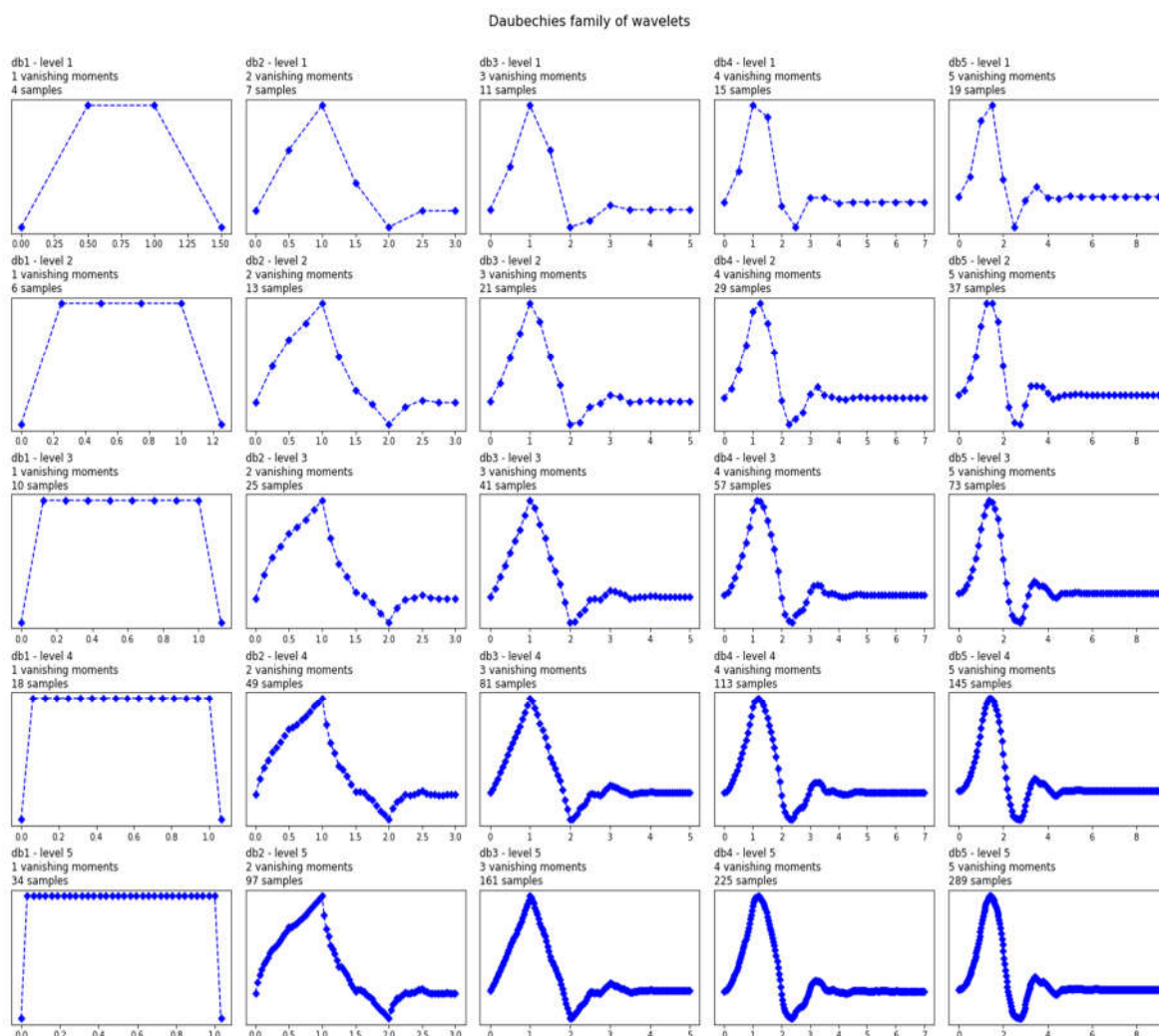
یک موجک می‌تواند متقارن (Symmetric) یا غیرمتقارن باشد.

یک موجک می‌تواند حقیقی یا مختلط باشد. موجک همیشه به دو قسمت تقسیم می‌شود، قسمت حقیقی که نشان‌دهنده دامنه و قسمت موهومی که نشان‌دهنده فاز است.

یک موجک نرمالیزه می‌شود تا دارای انرژی واحد باشد.

با استفاده از قطعه کد زیر می‌توانیم تصویری از خانواده‌های مختلف موجک را ترسیم کنیم. ردیف اول شامل چهار نوع موجک گسسته و ردیف دوم شامل چهار نوع موجک پیوسته است.

Python خروجی این کد به صورت زیر است. در هر یک از خانواده‌های موجک، زیرگروه‌های موجک مختلفی نیز وجود دارد که متعلق به آن خانواده است. برای تشخیص زیرگروه‌های مختلف در هر خانواده از موجک، باید به تعداد ضرایب (Vanishing Moments) و سطح تجزیه (Level of Decomposition) توجه کنیم. این مفهوم در کد زیر برای یک خانواده از موجک‌ها با نام Daubechies نشان داده شده است.



در تصویر بالا، می‌توانیم تبدیل موجک خانواده Daubechies یا به اختصار db را مشاهده کنیم. در ستون اول، تبدیل موجک db مرتبه (Order) اول (db1)، در ستون دوم تبدیل db2 و به همین ترتیب تا در ستون پنجم تبدیل موجک مرتبه ۵ ترسیم شده است. کتابخانه PyWavelets شامل موجک Daubechies تا مرتبه ۲۰ است. تعداد مرتبه‌های تبدیل موجک نشان‌دهنده تعداد لحظات محوشدگی (Vanishing Moments) است. بنابراین db3 دارای سه لحظه محوشدگی و db5 دارای ۵ لحظه محوشدگی است. تعداد لحظات محوشدگی به درجه تقریب و همواری (Smoothness) تبدیل موجک بستگی دارد. اگر یک تبدیل موجک دارای p لحظه محوشدگی باشد، آن‌گاه می‌تواند یک تابع چندجمله‌ای از درجه $p-1$ را تقریب بزند. هنگام انتخاب کردن تبدیل موجک، می‌توانیم تعیین کنیم که درجه تجزیه (Level of Decomposition) باید چقدر باشد. به صورت پیش فرض کتابخانه PyWavelets بیشینه درجه تجزیه ممکن را برای سیگنال ورودی انتخاب می‌کند. بیشینه درجه تجزیه توسط دستور `pywt.dwt_max_level` مشخص می‌شود و به طول سیگنال ورودی و موجک بستگی دارد. واضح است که هر چقدر تعداد لحظات محوشدگی افزایش یابد، درجه چندجمله‌ای موجک افزایش

می‌یابد و هموارتر (نرم‌تر) می‌شود و هرچه درجه تجزیه افزایش یابد، تعداد نمونه‌هایی که موجک بر اساس آن بیان می‌شود، افزایش می‌یابد.

تبدیل موجک پیوسته و تبدیل موجک گسسته

همان طور که در تصاویر دیدیم، تبدیل موجک دارای دو نوع مختلف گسسته و پیوسته است. از لحاظ ریاضی، یک تبدیل موجک پیوسته را می‌توان توسط تابع زیر توصیف کرد:

$$X\omega(a,b)=1|a|^{1/2}\int_{-\infty}^{\infty}x(t)^{-}\psi(t-ba)dt \quad X\omega(a,b)=1|a|^{1/2}\int_{-\infty}^{\infty}x(t)\psi^{-}(t-ba)dt$$

در فرمول بالا، $\psi(t)\psi(t)$ موجک مادر پیوسته است که توسط فاکتور aa مقیاس و توسط فاکتور bb منتقل

شده است. مقدار فاکتور مقیاس‌دهی و جابه‌جایی اعدادی پیوسته هستند، بنابراین تعداد بی شماری موجک

وجود دارد. می‌توان موجک مادر را توسط فاکتور 10^3 یا 10^31111 یا هر عدد دیگر مقیاس کرد.

اما زمانی که راجع به تبدیل موجک گسسته بحث می‌کنیم، تفاوت اصلی که وجود دارد این است که از مقادیر گسسته برای فاکتور مقیاس‌دهی و جابه‌جایی در این تبدیل استفاده می‌شود. فاکتور مقیاس به صورت اعداد توان

دو افزایش می‌یابد، بنابراین $a=1,2,4,...$ $a=1,2,4,...$ خواهد بود. فاکتور جابه‌جایی به صورت اعداد صحیح

افزایش می‌یابند، بنابراین $b=1,2,3,...$ $b=1,2,3,...$ است. نکته مهمی که وجود دارد این است که تبدیل

موجک گسسته فقط در دو حوزه مقیاس و جابه‌جایی گسسته است و در حوزه زمان گسسته نیست. برای کار

کردن با سیگنال‌های دیجیتال و گسسته، لازم است که تابع موجک را در حوزه زمان گسسته‌سازی کنیم. این

فرم‌های تبدیل موجک را تبدیل موجک گسسته در زمان یا تبدیل موجک پیوسته گسسته در زمان می‌گویند.

تبدیل فوریه گسسته به صورت بانک فیلتری در عمل، تبدیل فوریه گسسته به صورت یک بانک فیلتری

پیاده‌سازی می‌شود، بدین معنی که به صورت دنباله‌ای از فیلترهای پایین‌گذر و بالا‌گذر عمل می‌کند، به این

دلیل که استفاده از بانک فیلتری یک راه بسیار موثر برای تجزیه یک سیگنال به زیرباندهای فرکانسی

(Frequency Sub-bands) متعدد است. تلاش داریم تا در ادامه مفهوم پنهان در بانک فیلتری را به صورت

ساده توضیح دهیم. درک عملکرد بانک فیلتری برای درک عملکرد تبدیل موجک ضروری است و می‌تواند در

کاربردهای مختلف مورد استفاده قرار گیرد. برای اعمال تبدیل فوریه گسسته بر روی یک سیگنال، ابتدا با

مقیاس‌های کوچک شروع می‌کنیم. همان طور که قبلاً گفتیم، مقیاس‌های کوچک متناظر با فرکانس‌های بالا

است. بنابراین، ما ابتدا فرکانس‌های بالا را آنالیز می‌کنیم. در گام دوم، مقیاس را با فاکتور دو افزایش می‌دهیم

(فرکانس را با فاکتور دو کاهش می‌دهیم). و در این حالت رفتار را در اطراف نصف فرکانس بیشینه آنالیز

می‌کنیم. در گام سوم، فاکتور مقیاس را برابر با ۴ در نظر می‌گیریم و ما رفتار فرکانسی را در اطراف ربع فرکانس

بیشینه تحلیل می‌کنیم. این روال به همین ترتیب ادامه می‌یابد تا به بیشینه سطح تجزیه برسیم. برای درک

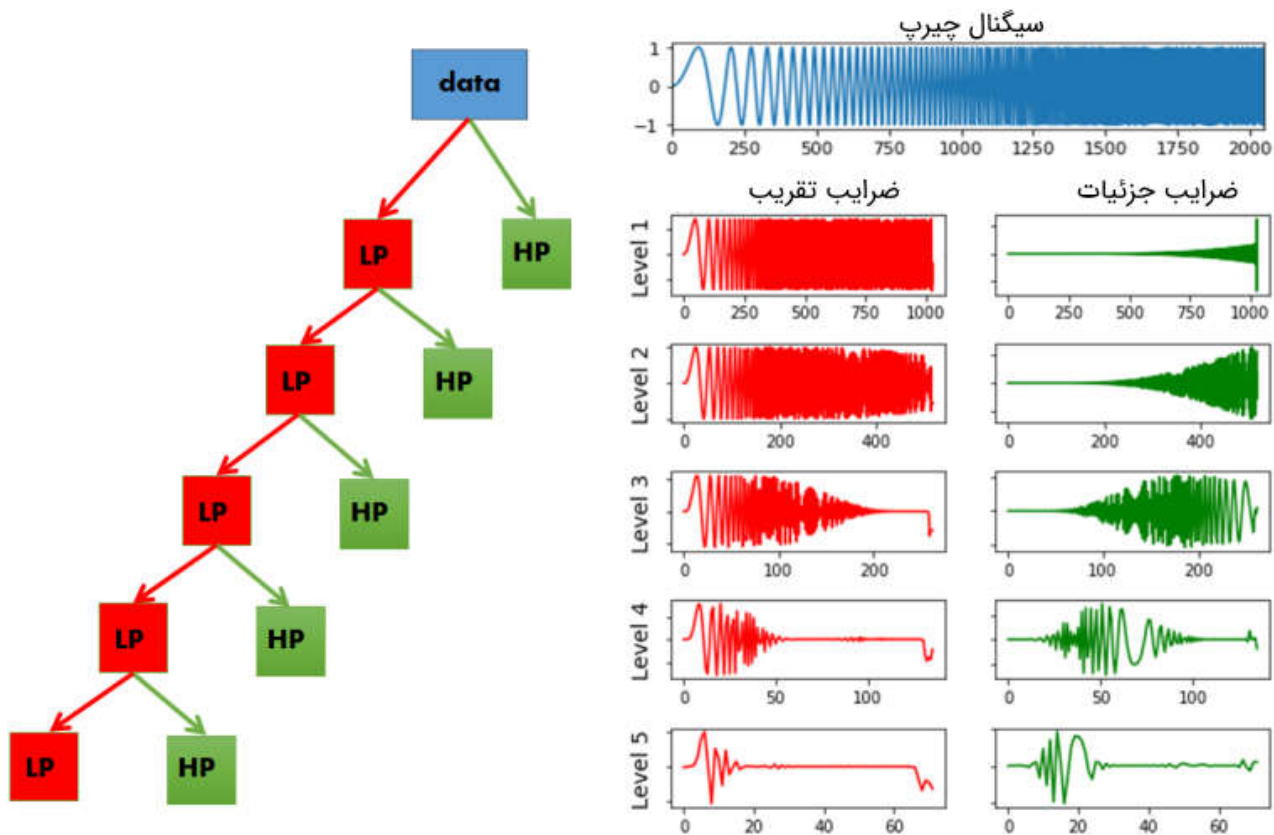
مفهوم بیشینه سطح تجزیه (Maximum Decomposition Level)، ابتدا باید بدانیم که در هر مرحله

متوالی، تعداد نمونه‌ها در سیگنال با فاکتور دو کاهش می‌یابد. در مقادیر فرکانس‌های پایین، به نمونه‌های

کمتری نیاز داریم تا در نرخ نمونه‌برداری نایکوییست صدق کنند، بنابراین لازم نیست تعداد نمونه‌های بیشتری را

در سیگنال حفظ کنیم؛ زیرا این کار تنها باعث می‌شود که هزینه محاسباتی بالاتر برود. به دلیل زیرنمونه‌برداری

(Downsampling) در بعضی از گام‌های فرایند، تعداد نمونه‌ها در سیگنال از طول فیلتر موجک کوچک‌تر می‌شود و در این حالت به بیشینه سطح تجزیه می‌رسیم. به عنوان مثال، فرض کنید که سیگنالی با فرکانس‌های در بازه 0 تا ۱۰۰۰ هرتز داریم. در گام اول، سیگنال را به دو بخش فرکانس بالا و فرکانس پایین تجزیه می‌کنیم. مثلاً بخش اول فرکانس‌های ۰ تا ۵۰۰ هرتز و بخش دوم فرکانس‌های ۵۰۰ تا ۱۰۰۰ هرتز هستند. در گام دوم، بخش فرکانس پایین را در نظر می‌گیریم و مجدداً آن را به دو بخش با فرکانس‌های ۰ تا ۲۵۰ هرتز و ۲۵۰ تا ۵۰۰ هرتز تقسیم می‌کنیم. در گام سوم، بخش ۰ تا ۲۵۰ هرتز را در نظر می‌گیریم و آن را به دو بخش ۰ تا ۱۲۵ هرتز و ۱۲۵ تا ۲۵۰ هرتز تقسیم می‌کنیم. این روند را به همین صورت تا زمانی ادامه می‌دهیم که یا به درجه پالایش مورد نیاز برسیم و یا تعداد نمونه‌ها تمام شود. می‌توان این ایده را به سادگی با استفاده از قطعه کد زیر به صورت تصویری نشان داد. در این کد نشان می‌دهیم که با اعمال تبدیل موجک گسسته روی یک سیگنال چیرپ (Chirp Signal)، چه اتفاقی رخ می‌دهد. سیگنال چیرپ، سیگنالی است که دارای طیف فرکانسی دینامیک باشد. در واقع طیف فرکانسی با زمان افزایش می‌یابد، سیگنال در ابتدا دارای مقادیر فرکانس پایین است و هر چه به سمت آخر سیگنال پیش می‌رویم، مقادیر سیگنال فرکانس بالاتر می‌شوند. استفاده از سیگنال چیرپ، مصورسازی این که کدام بخش از طیف فرکانسی فیلتر شده است را از طریق نگاه کردن به محور زمان، ساده‌تر می‌کند. در تصویر زیر نمایی از نحوه عملکرد تبدیل موجک گسسته به عنوان بانک فیلتری و خروجی قطعه کد بالا را می‌توان دید.



نحوه عملکرد تبدیل موجک گسسته به عنوان بانک فیلتری

در تصویر بالا می‌توان سیگنال چیرپ و تبدیل موجک گسسته اعمال شده به آن را مشاهده کرد. در رابطه با این تبدیل ذکر چند نکته اهمیت بالایی دارد. در کتابخانه PyWavelets، تبدیل موجک گسسته توسط دستور `pywt.dwt()` اعمال می‌شود. تبدیل فوریه گسسته دو مجموعه از ضرایب را به عنوان خروجی باز می‌گرداند، ضرایب تقریب (Approximation Coefficients) و ضرایب جزئیات (Detail Coefficients). ضرایب تقریب نشان‌دهنده خروجی فیلتر پایین‌گذر (فیلتر میانگین‌گیر) در تبدیل موجک گسسته هستند. ضرایب جزئیات نشان‌دهنده خروجی فیلتر بالاگذر (فیلتر مشتق‌گیر) در تبدیل موجک گسسته هستند. با اعمال تبدیل فوریه گسسته مجدداً روی ضرایب تقریب تبدیل موجک قبلی، تبدیل موجک مربوط به مرحله بعد را به دست می‌آوریم. در هر گام، نمونه‌برداری سیگنال اصلی با فاکتور دو کاهش می‌یابد. حال می‌توان درک کرد که چرا تبدیل موجک گسسته به صورت بانک فیلتری پیاده‌سازی می‌شود. در هر مرحله متوالی، ضرایب تقریب به دو بخش پایین‌گذر و بالاگذر تقسیم می‌شوند و در مرحله بعد، تبدیل موجک مجدداً روی بخش پایین‌گذر اعمال می‌شود. همان‌طور که می‌توان دید، سیگنال اصلی ما در حال حاضر به سیگنال‌های متعددی تبدیل شده است که هر کدام متناظر با باندهای فرکانسی مختلفی هستند. بعداً خواهیم دید که ضرایب تقریب و جزئی در زیر باندهای مختلف چگونه در کاربردهایی مانند حذف نویزهای فرکانس بالا از سیگنال‌ها، فشرده کردن سیگنال‌ها و یا طبقه‌بندی (Classifying) انواع مختلف سیگنال‌ها مورد استفاده قرار می‌گیرند. همچنین می‌توانیم از دستور `pywt.wavedec` برای محاسبه سریع ضرایب از مرتبه بالاتر استفاده کرد. این دستور تابع اصلی و مرتبه n را به عنوان ورودی دریافت می‌کند و یک مجموعه از ضرایب تقریب (از مرتبه n) و n مجموعه از ضرایب جزئیات (از مرتبه یک تا n) را به عنوان خروجی باز می‌گرداند. ایده آنالیز سیگنال با مقیاس‌های متفاوت را آنالیز چند-رزولوشنی (Multiresolution) یا چند-مقیاسی (Multiscale) می‌گویند و تجزیه سیگنال به این روش را نیز تجزیه چندرزولوشنی یا کدینگ زیرباند (Sub-Band Coding) می‌گویند.

مصورسازی فضای حالت با استفاده از تبدیل موجک پیوسته

تا این قسمت با مفهوم تبدیل موجک، تفاوت آن با تبدیل فوریه، تفاوت بین تبدیل موجک پیوسته و تبدیل موجک گسسته، انواع خانواده‌های مختلف تبدیل موجک، تاثیر درجه و سطح تجزیه روی موجک مادر آشنا شدیم و دانستیم چرا و چگونه تبدیل موجک گسسته مانند بانک فیلتری عمل می‌کند. همچنین دیدیم که خروجی تبدیل موجک روی سیگنال یک بعدی، منتج به یک اسپکتروگرام دو بعدی می‌شود. این اسپکتروگرام می‌تواند اطلاعات جزئی را درباره فضای حالت سیستم در اختیار ما قرار دهد یا به عبارت دیگر اطلاعاتی راجع به رفتار دینامیک سیستم در خود دارد.

دیتاست el-Nino یک سری زمانی است که برای ردیابی پدیده ال نینو مورد استفاده قرار می‌گیرد. این دیتاست شامل اندازه گیری‌های سه ماهه دمای سطح دریا از سال 1871 تا 1997 است. به منظور درک قدرت یک اسپکتروگرام، می‌خواهیم تبدیل موجک را برای داده‌های دیتاست ال نینو همراه با داده‌های سری زمانی واقعی و تبدیل فوریه آن‌ها نشان دهیم. این کار را با استفاده از قطعه کد زیر انجام می‌دهیم. نتیجه اجرای این کد تصاویر

زیر است که می‌توانید آن‌ها را مشاهده کنید. با دقت در تصویر بالا می‌توان دید که در اولین نمودار، سری زمانی دیتاست ال نینو به همراه میانگین زمانی آن نشان داده شده است. در تصویر وسط، تبدیل فوریه سری زمانی و در تصویر سوم، اسپکتروگرام این سیگنال را که از طریق تبدیل موجک پیوسته سیگنال اصلی به دست آمده است، می‌توان مشاهده کرد. در این اسپکتروگرام می‌توان دید که قسمت عمده توان در یک دوره ۲ تا ۸ ساله متمرکز شده است. اگر این دوره را از طریق فرمول $T=1fT=1f$ به فرکانس تبدیل کنیم، آن‌گاه به بازه فرکانس ۰.۱۲۵ هرتز تا ۰.۵ هرتز خواهیم رسید. البته از روی نمودار تبدیل فوریه نیز می‌توان افزایش توان در اطراف این فرکانس‌ها را مشاهده کرد. اما تفاوت اصلی که نمودار تبدیل فوریه و نمودار تبدیل موجک با یکدیگر دارند، در این است که تبدیل موجک می‌تواند اطلاعات زمانی را نیز در اختیار ما قرار دهد، در حالی که نمودار تبدیل فوریه این توانایی را ندارد. به عنوان مثال، در اسپکتروگرام می‌توان دید که تا سال ۱۹۲۰ نوسانات بسیاری در تغییرات دما وجود دارد، در حالی که بین سال‌های ۱۹۶۰ تا ۱۹۹۰ نوسانات به این شدت نیستند. همچنین می‌توان دید که با پیشرفت زمان، یک انتقال از تناوب‌های کوتاه به تناوب‌های بلند به وجود آمده است. در واقع، این یک نوع رفتار دینامیک در سیستم است که می‌توان با استفاده از تبدیل موجک به وجود آن پی برد، در حالی که تبدیل فوریه قادر به آشکارسازی رفتارهای دینامیک یک سیگنال نخواهد بود. اکنون واضح‌تر شده است که تبدیل موجک در کاربردهای پردازش سیگنال و یادگیری ماشین تا چه اندازه قدرتمندتر عمل می‌کند. برای کامل‌تر شدن بحث، می‌خواهیم به نحوه کاربرد تبدیل موجک به همراه شبکه عصبی کانولوشنی در طبقه‌بندی یک سیگنال بپردازیم.

استفاده از تبدیل موجک در طبقه‌بندی سیگنال

همان‌طور که گفتیم، تبدیل موجک یک سیگنال یک بعدی، یک اسپکتروگرام دو بعدی خواهد بود که حاوی اطلاعات بیشتری نسبت به سیگنال سری زمانی و یا تبدیل فوریه آن است. این حقیقت را با اعمال تبدیل موجک بر روی دیتاست ال نینو مشاهده کردیم. با اعمال تبدیل موجک روی این دیتاست، نه تنها به اطلاعات دوره تناوب بزرگترین نوسانات در دمای سطح دریا پی می‌بریم، بلکه خواهیم دانست این نوسانات چه زمانی رخ داده‌اند و چه زمانی وجود نداشته‌اند. این اسپکتروگرام، برای درک بهتر رفتار دینامیک یک سیگنال مورد استفاده قرار می‌گیرد و نیز با استفاده از آن می‌توان انواع مختلف سیگنال‌های تولید شده توسط یک سیستم را از یکدیگر تمایز داد. اگر یک سیگنال حیاتی بدن را هنگام بالا رفتن از پله‌ها و نیز پایین رفتن از پله‌ها ضبط کنید، اسپکتروگرام سیگنال در این دو حالت با یکدیگر تفاوت خواهند داشت. به همین ترتیب سیگنال‌های الکتروکاردیوگرافی (ECG) افراد سالم با سیگنال‌های ECG افراد آریتمی مختلف خواهد بود. از این ویژگی می‌توان در تشخیص عیب (Fault Detection) ادوات و دستگاه‌های مکانیکی مانند موتورهای و توربین‌های بادی نیز می‌توان استفاده کرد. بنابراین با نگاه کردن به اسپکتروگرام، می‌توان یک روتور موتور معیوب را از یک روتور سالم یا یک فرد سالم از فرد بیمار و یا بالا رفتن از پله را از پایین رفتن از آن تشخیص داد. اما گاهی لازم است که این کار را بدون چک کردن دستی اسپکتروگرام‌های متعدد مربوط به موارد مختلف انجام دهیم. یک راه برای انجام اتوماتیک این کار، ایجاد یک شبکه عصبی کانولوشنی است. این شبکه عصبی قادر است که به صورت

خودکار، کلاسی که اسپکتروگرام به آن تعلق دارد را مشخص کند و در نتیجه سیگنال‌ها را بر اساس آن طبقه‌بندی کند. اگر با شبکه عصبی کانولوشن آشنا نیستید، بهتر است که ابتدا مطلب «بررسی شبکه عصبی کانولوشن (CNN)» را در مجله فرادرس مطالعه کنید. در بخش‌های بعد، ابتدا یک دیتاست را بارگذاری می‌کنیم که اطلاعات افراد در حین انجام شش فعالیت مختلف را ضبط کرده است. سپس اسپکتروگرام این دیتاست را با استفاده از تبدیل موجک ترسیم می‌کنیم و در نهایت با استفاده از شبکه عصبی کانولوشنی این سیگنال‌ها را طبقه‌بندی خواهیم کرد.

بارگذاری دیتاست

حال می‌خواهیم داده‌های یک دیتاست شامل سری زمان را با استفاده از اسپکتروگرام و شبکه CNN طبقه‌بندی کنیم. ابتدا دیتاست تشخیص فعالیت‌های انسان (UCI-HAR) را از این لینک دانلود و سپس بارگذاری می‌کنیم. این دیتاست با استفاده از سنسورها از افراد در حال انجام کارهای مختلف مانند بالا و پایین رفتن از پله، خوابیدن، ایستادن و راه رفتن ضبط شده است. در کل، 10000 سیگنال در این دیتاست قرار دارد که هر کدام از آن‌ها از ۹ المان تشکیل شده‌اند. بعد از اینکه دیتاست را دانلود کردیم، می‌توانیم آن را با کد زیر در یک nd-array از کتابخانه numpy بارگذاری کرد.

داده‌های آموزش (Training Set) شامل 7352 سیگنال هستند که هر سیگنال ۱۲۸ نمونه اندازه‌گیری و 9 المان دارد. بنابراین داده‌های آموزش در یک ndarray با ابعاد (7352,128,9) و داده‌های تست (Test Set) در یک آرایه دیگر با ابعاد (2947,128,9) بارگذاری می‌کنیم. چون هر سیگنال از ۹ المان تشکیل شده است، در نتیجه باید تبدیل موجک پیوسته را به هر سیگنال ۹ بار اعمال کنیم. در تصویر زیر نتیجه حاصل از اعمال تبدیل موجک پیوسته به دو سیگنال مختلف از دیتاست را می‌توان مشاهده کرد.

در تصویر بالا، نمونه‌های حاصل از اعمال تبدیل موجک به یک سیگنال فرد در حال بالا رفتن از پله‌ها و فرد خوابیده را می‌توان مشاهده کرد.

اعمال تبدیل موجک پیوسته بر روی دیتاست

چون هر سیگنال دارای ۹ مولفه است، بنابراین هر سیگنال ۹ اسپکتروگرام نیز دارد. بنابراین سوال بعدی این است که چگونه این ۹ اسپکتروگرام را به شبکه کانولوشنی وارد کنیم. راه‌های مختلفی برای این کار وجود دارد. راه اول این است که برای هر مولفه، یک شبکه CNN را به صورت جداگانه آموزش دهیم و سپس نتایج این ۹ شبکه را از طریق یکی از روش‌های Ensembling با یکدیگر مخلوط کنیم. اما این روش در حالت کلی منجر به نتایج ضعیفی می‌شود؛ زیرا وابستگی‌های داخلی بین این ۹ مولفه را در شبکه عصبی دخالت نداده‌ایم. راه دوم الحاق کردن (Concatenate) این ۹ سیگنال مختلف به یکدیگر و ایجاد یک سیگنال طولانی‌تر و سپس اعمال شبکه عصبی به سیگنال الحاقی است. این روش نیز می‌تواند کارا باشد، اما در محل الحاق سیگنال‌ها به یکدیگر، شاهد ناپیوستگی‌هایی خواهیم بود. این ناپیوستگی‌ها می‌توانند در محل مرزهای مولفه‌های سیگنال در اسپکتروگرام منجر به ایجاد نویز شوند. راه سوم این است که ابتدا تبدیل موجک پیوسته را محاسبه

کنیم و سپس ۹ تصویر تبدیل موجک مختلف را به یکدیگر الحاق کنیم و تصویر نهایی را به شبکه عصبی وارد کنیم. این روش نیز موثر است، اما در آن نیز مجدداً در لبه‌های تصاویر تبدیل موجک، ناپیوستگی به وجود می‌آید که باعث ایجاد نویز می‌شود و در نهایت وارد شبکه عصبی می‌شود. اگر شبکه عصبی کانولوشنی به اندازه کافی عمیق باشد، آن‌گاه قادر خواهد بود که تفاوت بین این قسمت‌های نویزی و قسمت‌های مفید و واقعی تصویر را تشخیص دهد. با این وجود، استفاده از راه حل چهارم پیشنهاد می‌شود.

هر ۹ اسپکتروگرام را روی (Top) یکدیگر قرار دهید و یک تصویر با ۹ کانال مختلف را ایجاد کنید. اما تصویری با ۹ کانال به چه معنی است؟ در حالت عادی، هر تصویر یا سیاه و سفید است و یک کانال دارد و یا تصویر از نوع رنگی (RGB) بوده و سه کانال دارد. اما شبکه عصبی کانولوشنی برای سادگی در مدیریت تصاویر می‌تواند این تصاویر مختلف را در کنار یکدیگر و به عنوان ۹ کانال در نظر بگیرد. طریقه عملکرد شبکه عصبی CNN همچنان مانند قبل است، اما تفاوتی که وجود دارد این است که در مقایسه با یک تصویر RGB، سه برابر فیلتر بیشتری وجود خواهد داشت. در تصویر زیر روند انجام این کار را می‌توان مشاهده کرد.

در ادامه، قطعه کد پایتون مربوط به اعمال تبدیل موجک پیوسته روی سیگنال‌های دیتاست و نیز بازآرایی مجدد آن‌ها برای کاربرد در ورودی شبکه عصبی کانولوشنی را مشاهده می‌کنید. دیتاست در حالت کلی شامل ۱۰۰۰۰ سیگنال است، اما در اینجا از ۵۰۰۰ سیگنال به عنوان داده آموزش و از ۵۰۰ سیگنال دیگر به عنوان داده تست استفاده می‌کنیم.

همان طور که از قطعه کد بالا مشخص است، تبدیل موجک یک مولفه تکی سیگنال (۱۲۸ نمونه) منجر به یک تصویر با ابعاد ۱۲۷ در ۱۲۷ پیکسل می‌شود، بنابراین اسپکتروگرام حاصل از ۵۰۰۰ داده آموزش، در یک آرایه با ابعاد (5000,127,127,9)(5000,127,127,9) ذخیره می‌شود. همچنین اسپکتروگرام حاصل از ۵۰۰ سیگنال تست، در آرایه numpy دیگری با ابعاد (500,127,127,9)(500,127,127,9) ذخیره‌سازی می‌شود.

آموزش شبکه عصبی کانولوشنی

حال که داده‌ها در فرمت صحیحی قرار گرفته‌اند، می‌توان شبکه عصبی CNN را آموزش داد. برای این قسمت لازم است تا از کتابخانه Keras استفاده کنیم. بنابراین باید ابتدا آن را نصب کنیم. در ادامه کدهای پایتون برای آموزش شبکه عصبی آورده شده‌اند: دقت طبقه‌بندی دیتاست UCI-HAR با استفاده از ترکیب شبکه عصبی کانولوشنی و تبدیل موجک را روی داده‌های آموزش و تست می‌توان در تصویر زیر مشاهده کرد.

همان طور که می‌توان دید، تلفیق تبدیل موجک و شبکه عصبی کانولوشنی منجر به نتایج فوق‌العاده‌ای می‌شود. در طبقه‌بندی انواع مختلف فعالیت‌های انسان، استفاده از این روش منجر به 94 درصد دقت (Accuracy) می‌شود. این دقت بسیار بالاتر از دقتی است که با سایر روش‌های طبقه‌بندی می‌توان به آن رسید. در بخش بعد، از تبدیل فوریه گسسته استفاده می‌کنیم و مجدداً داده‌های همین دیتاست را طبقه‌بندی می‌کنیم و نتایج را در دو حالت با یکدیگر مقایسه می‌کنیم.

تجزیه سیگنال با استفاده از تبدیل موجک گسسته

در بخش‌های قبلی دیدیم که چگونه تبدیل موجک گسسته مانند یک بانک فیلتری عمل می‌کند و می‌تواند یک سیگنال را به فرکانس‌های زیر باند تجزیه کند. در این بخش، می‌خواهیم نحوه تجزیه یک سیگنال به فرکانس‌های زیر باند و بازسازی مجدد سیگنال اصلی را با استفاده از کتابخانه PyWavelets در پایتون بررسی کنیم. PyWavelets دو روش اساسی برای تجزیه یک سیگنال در اختیار کاربر قرار می‌دهد.

در روش اول می‌توان `pywt.dwt()` را روی سیگنال اعمال کرد و ضرایب تقریب را بازیابی کرد. سپس باید تبدیل موجک گسسته را روی ضرایب بازیابی شده اعمال کرد تا ضرایب مرحله دوم را به دست آورد. این فرایند را باید تا رسیدن به سطح تجزیه مطلوب ادامه داد. در روش دوم، می‌توان تابع `pywt.wavedec()` را مستقیماً روی سیگنال اعمال کرد و تمام ضرایب جزئیات را تا سطح تجزیه دلخواه n به دست آورد. این تابع، سیگنال اصلی و سطح n را به عنوان ورودی دریافت می‌کند و یک مجموعه از ضرایب تقریب (مربوط به سطح تجزیه n) و n مجموعه از ضرایب جزئیات (از سطح یک تا سطح n) را در خروجی باز می‌گرداند. با استفاده از قطعه کد زیر می‌توان نتیجه استفاده از روش اول را بررسی کرد.

در تصویر بالا، یک سیگنال را به ضرایب آن تجزیه کردیم و مجدداً با استفاده از معکوس تبدیل موجک گسسته بازسازی نمودیم.

در روش دوم، از تابع `pywt.wavedec()` استفاده می‌شود تا یک سیگنال را تجزیه و سپس مجدداً بازسازی کرد. روش دوم ساده‌ترین راه برای تجزیه و بازسازی یک سیگنال است، مخصوصاً زمانی که بخواهید ضرایب مرتبه بالاتر را به دست آورید. قطعه کد زیر نحوه استفاده از این تابع را نشان می‌دهد. سیگنال اصلی را به همراه سیگنال بازسازی شده با روش دوم حذف نویز (مولفه‌های فرکانس بالا) با استفاده از تبدیل موجک گسسته

در بخش قبل دیدیم که چگونه می‌توان یک سیگنال را به ضرایب تقریب (پایین گذر) و ضرایب جزئیات (بالا گذر) تجزیه کرد. اگر سیگنال را با استفاده از این ضرایب بازسازی کنیم، در نهایت سیگنال اصلی را می‌توان مجدداً به دست آورد. اما اگر تعدادی از ضرایب جزئیات را از دست بدهیم، آن‌گاه در بازسازی سیگنال اصلی چه اتفاقی می‌افتد؟ به دلیل این که ضرایب جزئیات، نشان‌دهنده قسمت فرکانس بالا در سیگنال هستند، در نتیجه می‌توان گفت که در اثر حذف تعدادی از ضرایب جزئیات، بخشی از طیف فرکانسی فیلتر می‌شود. اگر در سیگنال نویزهای فرکانس بالای زیادی وجود داشته باشند، این روش می‌تواند به فیلتر و حذف کردن آن‌ها کمک کند. صرف نظر کردن از ضرایب جزئیات، با استفاده از دستور `pywt.threshold()` انجام می‌گیرد. در واقع این دستور مقادیر ضرایب بالاتر از حد آستانه (Threshold) خاصی را حذف می‌کند.

حال می‌خواهیم با استفاده از دیتاست «NASA's Femto Bearing Dataset» این مفهوم را نشان دهیم. این دیتاست شامل داده‌های فرکانس بالای سنسور در حضور عیب یا تاقان (Bearings) است. قطعه کد زیر را در محیط پایتون اجرا می‌کنیم. تصویر زیر در خروجی ایجاد می‌شود. در این تصویر سیگنال با مولفه‌های فرکانس بالا و سیگنال صاف‌شده آن پس از اعمال تبدیل موجک گسسته را می‌توان مشاهده کرد. سیگنال با مولفه‌های فرکانس بالا و سیگنال صاف‌شده آن پس از اعمال تبدیل موجک گسسته

همان طور که در تصویر بالا نیز مشاهده می‌شود، می‌توان از طریق تجزیه یک سیگنال و برابر صفر قرار دادن برخی از ضرایب در آن و سپس بازسازی مجدد سیگنال، نویزهای فرکانس بالا در سیگنال را حذف کرد. البته اگر با تکنیک‌های مختلف پردازش سیگنال آشنا باشیم، می‌دانیم که روش‌های مختلفی برای حذف نویز از یک سیگنال وجود دارد. به عنوان مثال، در پایتون کتابخانه Scipy دارای فیلترهای صاف‌کننده متنوعی است که استفاده از این فیلترها بسیار ساده‌تر است. یکی از فیلترهای صاف‌کننده، فیلتر Savitzky-Golay نام دارد. یکی دیگر از روش‌های حذف نویز، میانگین‌گیری از سیگنال در طول محور زمان است. سوالی که پیش می‌آید این است که چرا باید از روش تبدیل موجک گسسته که نسبتاً پیچیده‌تر است، به جای این روش‌ها استفاده کرد؟ در جواب باید گفت، مزیت اصلی تبدیل موجک گسسته از این ویژگی ناشی می‌شود که اشکال متنوعی از موجک وجود دارند که می‌توان از آن‌ها استفاده کرد. می‌توان یک موجک را انتخاب کرد که شکل آن متناظر با مشخصه‌هایی از پدیده باشد که انتظار مشاهده آن را داریم. به این طریق، تعداد کمتری از مشخصه‌های پدیده که مورد نیاز هستند، در اثر صاف شدن از بین می‌روند.

استفاده از تبدیل موجک گسسته برای طبقه‌بندی سیگنال در ادامه قصد داریم تا ببینیم چگونه می‌توان از تبدیل موجک گسسته و شبکه عصبی کانولوشنی برای طبقه‌بندی سیگنال‌ها استفاده کرد.

ایده پنهان در طبقه‌بندی سیگنال با استفاده از تبدیل موجک گسسته ایده‌ای است که در پس طبقه‌بندی سیگنال با استفاده از تبدیل موجک گسسته وجود دارد را می‌توان به صورت زیر بیان کرد: ابتدا از تبدیل موجک گسسته برای تفکیک یک سیگنال به زیرباندهای فرکانسی مختلف استفاده می‌کنیم. این کار را تا حد ممکن یا تا حد لازم تکرار می‌کنیم. چون انواع مختلف سیگنال‌ها، مشخصه‌های فرکانسی مختلفی را از خود نشان می‌دهند، در نتیجه این تمایز در رفتار باید در یکی از زیرباندهای فرکانسی خود را نشان دهد. بنابراین اگر ویژگی‌ها (Features) را از هریک از زیرباندهای مختلف ایجاد کنیم و از مجموعه تمام ویژگی‌ها به عنوان ورودی در یک طبقه‌بند (Classifier) مانند جنگل تصادفی، ارتقای گرادین و یا رگرسیون منطقی استفاده کنیم و الگوریتم طبقه‌بندی را با استفاده از این ویژگی‌ها آموزش دهیم، آن‌گاه الگوریتم قادر خواهد بود که تمایز بین سیگنال‌های مختلف را تشخیص دهد و طبقه‌بندی را بر اساس آن انجام دهد. این مفهوم در تصویر زیر به خوبی نشان داده شده است.

تولید ویژگی از هر زیر باند

حال می‌خواهیم بدانیم از مجموعه مقادیر هر زیر باند چه ویژگی‌هایی را می‌توان ایجاد کرد؟ واضح است که این امر تا حد بسیار زیادی به نوع سیگنال و کاربرد مورد نظر بستگی دارد. اما در حالت کلی، ویژگی‌های زیر را می‌توان به عنوان متداول‌ترین و پر تکرارترین ویژگی‌های مورد استفاده در سیگنال‌ها بیان کرد:

مقادیر ضرایب مدل اتورگرسیو (Auto-regressive) مقادیر آنتروپی (Entropy). این مقادیر را می‌توان به عنوان معیاری از پیچیدگی (Complexity) سیگنال در نظر گرفت.

و یا ویژگی‌های آماری مانند:

واریانس

انحراف معیار

میانگین، میانه

مقدار صدک 25ام

مقدار صدک 75ام

مقدار مجذور میانگین مربعات

میانگین مشتقات

نرخ عبور از صفر. (تعداد دفعات عبور سیگنال از $y=0$)

نرخ عبور از میانگین. (تعداد دفعات عبور سیگنال از $y=\text{mean}(y)$)

البته ایده دیگری نیز برای ایجاد ویژگی از زیرباندهای سیگنال وجود دارد. می‌توانید از تعدادی از ویژگی‌هایی که در اینجا توصیف شدند استفاده کنید و یا حتی می‌توانید تمامی آن‌ها را با هم به کار ببرید. اکثر طبقه‌بندها یا کلاسیفایرها در پکیج Scikit-Learn آنقدر توانمند هستند که می‌توانند تعداد زیادی ویژگی ورودی را مدیریت کنند و انواع مفید و غیر مفید ویژگی‌ها را از یکدیگر تمایز دهند. البته توجه کنید که تعداد بسیار زیادی توابع آماری در کتابخانه scipy.stats وجود دارند که می‌توان با استفاده از آن‌ها ویژگی‌های بیشتری را نیز ایجاد کرد. با استفاده از قطعه کد زیر می‌توان نحوه استفاده از تعدادی از ویژگی‌های بیان شده در بالا را مشاهده کرد. در قطعه کد بالا توابع زیر مورد استفاده قرار گرفته‌اند:

یک تابع برای محاسبه مقدار آنتروپی مربوط به سیگنال ورودی.

یک تابع برای محاسبه چند مشخصه آماری مانند: میانگین، چند صدک، انحراف معیار و واریانس و ...

یک تابع برای محاسبه نرخ عبور از صفر و نرخ عبور از میانگین.

یک تابع برای ترکیب نتایج سه تابع بالا.

تابع نهایی در برنامه بالا ۱۲ ویژگی متمایز را برای هر لیست از مقادیر در خروجی باز می‌گرداند. بنابراین اگر یک سیگنال به ۱۰ زیرباند متمایز تجزیه شود و برای هر زیرباند ویژگی‌هایی تولید کنیم، در نهایت 120 ویژگی برای هر سیگنال خواهیم داشت.

استفاده از ویژگی‌ها و کتابخانه Scikit-Learn برای طبقه‌بندی دو دیتاست ECG

تا این قسمت مطالب بسیاری را بیان کردیم. در گام بعد، باید سیگنال‌های مجموعه آموزش را با استفاده از تبدیل موجک گسسته به زیرباندهای خود تجزیه کنیم و برای هر زیرباند ویژگی‌ها را محاسبه کنیم. سپس از ویژگی‌ها برای آموزش یک طبقه‌بند استفاده کنیم تا طبقه‌بند بتواند سیگنال‌های مجموعه تست را پیش‌بینی کند. این فرایند را برای دو دیتاست سری زمانی انجام می‌دهیم. اولین دیتاست UCI-HAR است که در بخش‌های قبل از آن استفاده کردیم. این دیتاست شامل داده‌های ضبط شده سنسورهای گوشی‌های هوشمند از فعالیت‌های مختلف انسان مانند نشستن، ایستادن، راه رفتن، بالا رفتن از پله و پایین رفتن از پله است. دومین دیتاست، PhysioNet نام دارد که شامل داده‌های ECG است. دیتاست PhysioNet را می‌توانید با کلیک روی

لینک دانلود کنید. دیتاست PhysioNet شامل مجموعه‌ای از سیگنال‌های ECG اندازه‌گیری شده از افراد سالم (NSR) و افراد با بیماری ARR یا CHF است. در مجموع دیتاست از 96 اندازه‌گیری ARR و 36 اندازه‌گیری NSR و 30 اندازه‌گیری CHF تشکیل شده است. بعد از این که هر دو دیتاست را دانلود کردیم و آن‌ها را در پوشه صحیح قرار دادیم، گام بعدی این است که آن‌ها را در حافظه بارگذاری کنیم. در بخش قبل با نحوه بارگذاری دیتاست UCI-HAR آشنا شدیم. در قطعه کد زیر نحوه بارگذاری دیتاست ECG را مشاهده می‌کنید. توجه کنید که دیتاست ECG به صورت یک فایل MATLAB ذخیره شده است. بنابراین باید از `scipy.io.loadmat()` استفاده کنیم تا این فایل در پایتون باز شود و داده‌های اندازه‌گیری و برچسب هر کدام در دو لیست جداگانه بازبایی شود. توجه کنید که دیتاست UCI HAR در فایل‌هایی با پسوند `.txt` ذخیره شده است و بعد از خواندن داده‌ها، باید آن‌ها را در یک آرایه `numpy` با سایز `(no of signals, length of signal, no of components)=(10299,128,9)` (no of signals, length of signal, no of components)=(10299,128,9) ذخیره کنیم. حال می‌خواهیم ببینیم ویژگی‌های سیگنال‌ها به چه صورت از دیتاست‌ها استخراج می‌شوند. برای این کار باید قطعه کد زیر را در پایتون اجرا کنیم.

کاری که در کد بالا انجام دادیم این است که توابعی برای تولید ویژگی از سیگنال‌های ECG و UCI HAR نوشتیم. در واقع این دو تابع تفاوتی با یکدیگر ندارند و تنها دلیل اینکه از دو تابع مختلف برای این دو دیتاست استفاده کردیم این است که داده‌ها در هر کدام از آن‌ها با یک فرمت مختلف ذخیره شده است. داده‌ها در دیتاست ECG در قالب لیست ذخیره شده‌اند، در حالی که داده‌ها در دیتاست UCI HAR در قالب یک `numpy ndarray` سه بعدی ذخیره شده‌اند. در برنامه فوق برای داده‌های دیتاست ECG، روی لیست سیگنال‌ها حلقه زده‌ایم و روی هر سیگنال تبدیل موجک گسسته را اعمال کرده‌ایم که در نهایت لیستی از ضرایب را باز می‌گرداند. برای هر یک از این سیگنال‌ها یا به عبارت دیگر برای هر یک از این زیر باندها، ویژگی‌ها را با استفاده از تابعی که در بخش قبل تعریف کردیم، استخراج می‌کنیم. ویژگی‌هایی که با استفاده از تمام ضرایب مختلف مربوط به یک سیگنال محاسبه شده‌اند، به یکدیگر الحاق می‌شوند؛ زیرا این ویژگی‌ها در واقع متعلق به یک سیگنال هستند. همین کار را با داده‌های دیتاست UCI HAR نیز انجام می‌دهیم. تنها تفاوت در این است که این بار دو حلقه `for` داریم؛ زیرا هر سیگنال از ۹ مولفه تشکیل شده است. ویژگی‌هایی که از هر یک از زیر باندهای هر مولفه یک سیگنال تولید می‌شود، به یکدیگر الحاق می‌شوند. حال که ویژگی‌ها را برای هر دیتاست محاسبه کرده‌ایم، می‌توانیم از تابع دسته‌بند یا طبقه‌بند ارتقای گرادیان (`GradientBoostingClassifier`) در کتابخانه `scikit-learn` استفاده کنیم و آن را آموزش دهیم. برای این کار باید قطعه کد زیر را در پایتون اجرا کنیم. همان طور که می‌بینید، نتایج هنگام استفاده از تبدیل موجک گسسته به همراه `Gradient Boosting Classifier` بسیار عالی است. این روش روی دیتاست ECG دارای دقت 93 درصد و روی دیتاست UCI-HAR دارای دقت ۹۵ درصد است. مقایسه دقت طبقه‌بندی با استفاده از تبدیل موجک گسسته، تبدیل فوریه و شبکه عصبی بازگشتی (RNN) اگر طبقه‌بندی داده‌های دیتاست UCI-HAR را با استفاده از تبدیل فوریه انجام دهیم، آن‌گاه دقت بر روی داده‌های تست برابر با ۹۱ درصد خواهد بود.

از طرف دیگر، اگر طبقه‌بندی را با استفاده از شبکه عصبی بازگشتی انجام دهیم، آن‌گاه بالاترین دقت روی داده‌های تست حدوداً ۸۶ درصد به دست می‌آید. در حالی که با استفاده از تبدیل موجک و CNN توانستیم به دقت ۹۴ درصد و با استفاده از تبدیل موجک و ارتقای گرادیان به دقت ۹۵ درصد روی داده‌های تست برسیم. واضح است که استفاده از تبدیل موجک باعث شده که دقت طبقه‌بندی نسبت به سایر روش‌ها بسیار بالاتر باشد. این کار را می‌توان با دیتاست‌های دیگر نیز انجام داد و به احتمال زیاد باز هم نتایج با استفاده از تبدیل موجک نسبت به سایر روش‌ها بهتر خواهد بود. نکته دیگری که در اینجا لازم است به آن توجه کنیم این است که شبکه عصبی بازگشتی بدترین نتیجه را در بین این سه روش دارد. حتی استفاده از روش بسیار ساده‌ای مانند تبدیل فوریه و یافتن محل پیک‌های طیف فرکانسی، به نتیجه بهتری نسبت به استفاده از شبکه عصبی بازگشتی می‌توان دست یافت.

این نکته بسیار جالب است، زیرا یکی از ویژگی‌های شبکه عصبی بازگشتی، توانایی یادگیری وابستگی‌های موقتی (Temporal Dependencies) در داده‌های ترتیبی (Sequential Data) است. اگر تبدیل فوریه قادر است هر سیگنالی را بدون توجه به پیچیدگی آن توسط المان‌های فوریه توصیف کند، آن‌گاه با توجه به وابستگی‌های موقتی، چه چیزهای بیشتری را می‌تواند یاد بگیرد. دقت به دست آمده با استفاده از تبدیل موجک گسسته به ویژگی‌هایی که برای محاسبه در نظر گرفته می‌شود، موجک انتخاب شده و نوع الگوریتم دسته‌بند نیز بستگی دارد. برای یافتن درک نسبی، در تصویر زیر دقت طبقه‌بندی داده‌های تست روی دیتاست‌های UCI-HAR و ECG به ازای تمام موجک‌های PyWavelets و ۵ مورد از پرکاربردترین الگوریتم‌های طبقه‌بندی کتابخانه scikit-learn نشان داده شده است. با توجه به این تصویر می‌توان دید که دقت طبقه‌بندی، متناظر با الگوریتم طبقه‌بندی انتخاب شده و تابع موجک، تغییر خواهد کرد، اما در حالت کلی می‌توان گفت که الگوریتم ارتقای گرادیان عملکرد بهتری دارد. نکته دیگری که باید به آن اشاره کرد این است که موجک انتخاب شده می‌تواند تاثیر زیادی روی دقت نهایی طبقه‌بندی داشته باشد. بهترین راه برای انتخاب موجک مناسب، آزمایش و خطای بسیار و نیز انجام تحقیق روی کاربرد قبل از پیاده‌سازی است.

تبدیل موجک در متلب

برای آشنایی با پیاده‌سازی تبدیل موجک در متلب در این قسمت می‌خواهیم تبدیل موجک را روی بخشی از دیتاست ال نینو اعمال کنیم. داده‌ها را می‌توان از طریق کلیک روی `sst_nino3.dat` دانلود کرد. این فایل شامل داده‌های دیتاست ال نینو است که باید در قالب فایل `.dat` در متلب ذخیره شود. حال باید توابع `chisquare_solve.m` و `chisquare_inv.m` و `wave_bases.m` و `wave_signif.m` و `wavelet.m` را دانلود کرده و در یک محل قرار دهیم. در نهایت برای تست کردن نتیجه اعمال تبدیل موجک می‌توان کد زیر را اجرا کرد.

For any 2π periodical function $f(x)$:

$$f(x) = a_0 + \sum_{k=1}^{\infty} (a_k \cos kx + b_k \sin kx)$$

$$a_0 = \frac{1}{2\pi} \int_0^{2\pi} f(x) dx$$

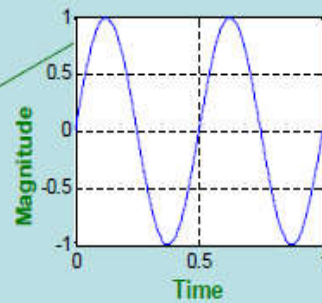
$$a_k = \frac{1}{\pi} \int_0^{2\pi} f(x) \cos(kx) dx$$

$$b_k = \frac{1}{\pi} \int_0^{2\pi} f(x) \sin(kx) dx$$

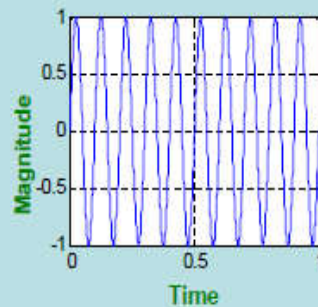


Jean-Baptiste-Joseph Fourier
(1768-1830)

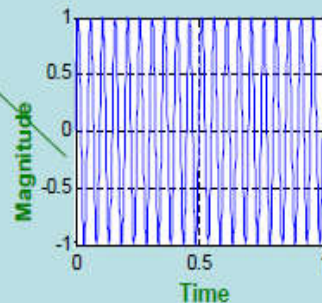
2 Hz



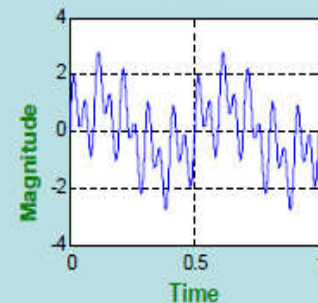
10 Hz



20 Hz

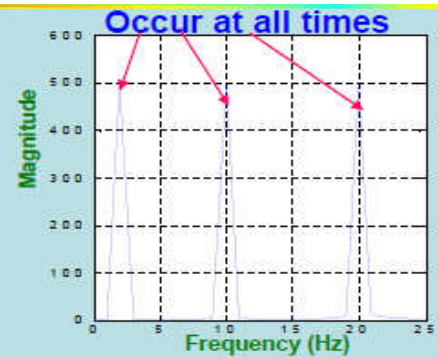
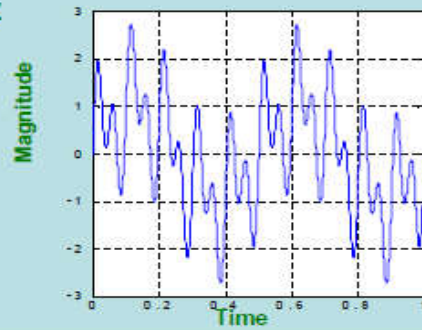


2 Hz +
10 Hz +
20Hz



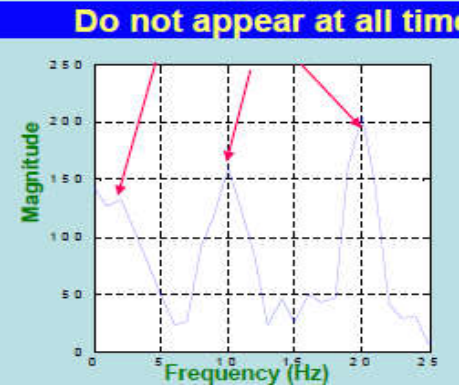
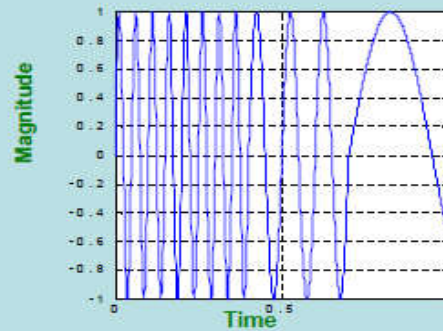
2 Hz + 10 Hz + 20Hz

Stationary



0.0-0.4: 2 Hz +
0.4-0.7: 10 Hz +
0.7-1.0: 20Hz

Non-Stationary

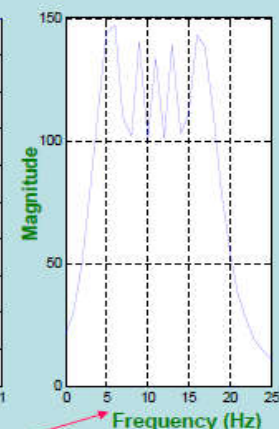
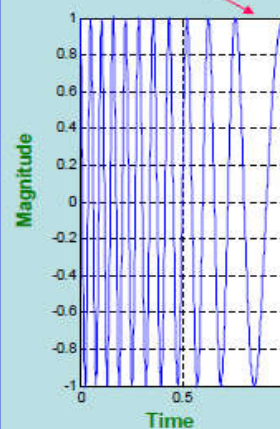
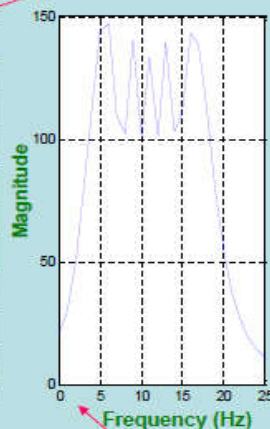
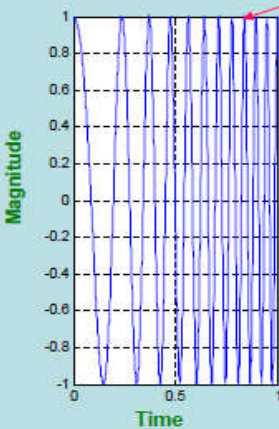


CHIRP SIGNALS

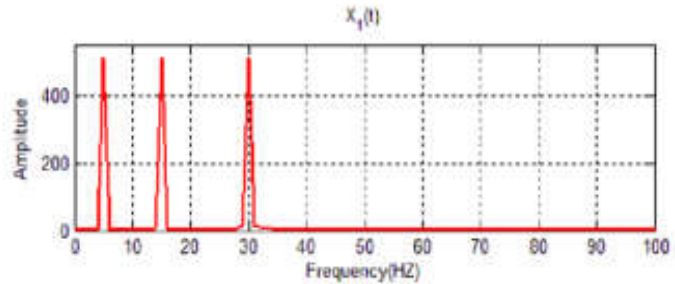
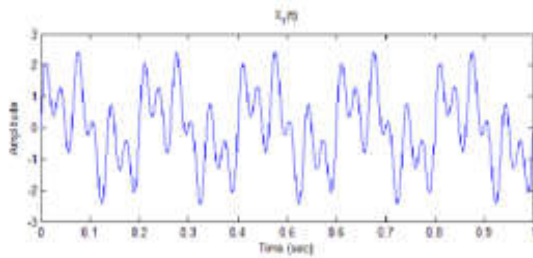
Frequency: 2 Hz to 20 Hz

Frequency: 20 Hz to 2 Hz

Different in Time Domain

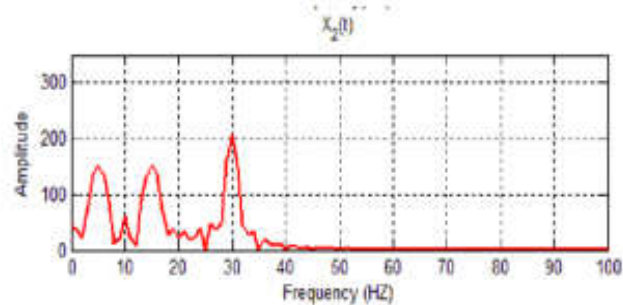
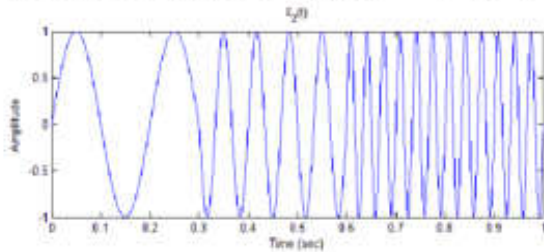


Same in Frequency Domain



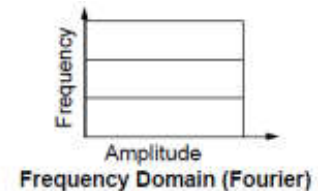
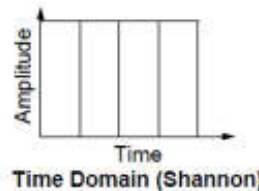
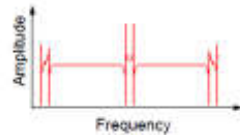
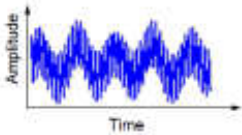
سیگنال ایستا $x_1(t) = \sin(2\pi 5t) + \sin(2\pi 15t) + \sin(2\pi 30t)$

تبدیل فوریه سیگنال ایستا

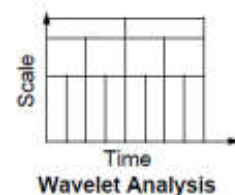
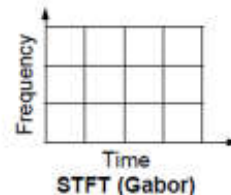
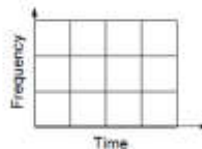
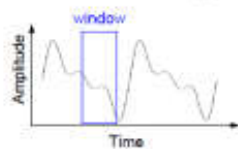


سیگنال غیر ایستا $x_2(t) = \begin{cases} \sin(2\pi 5t) & t < 0.3 \\ \sin(2\pi 15t) & 0.3 < t < 0.6 \\ \sin(2\pi 30t) & t > 0.6 \end{cases}$

تبدیل فوریه سیگنال غیر ایستا



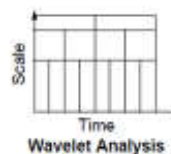
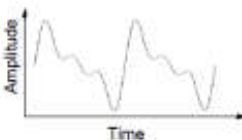
Fourier Analysis



Short-Time Fourier Analysis

$$Gf(s, t) = \int_{-\infty}^{\infty} f(u)g(u-t)e^{-2\pi i s t} du$$

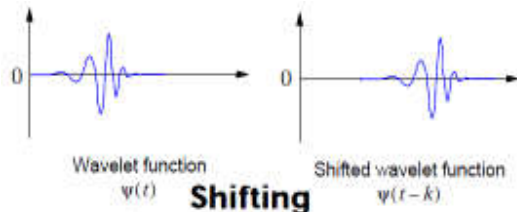
$$F(\omega) = \int_{-\infty}^{\infty} f(t)e^{-j\omega t} dt$$



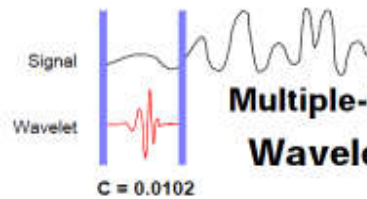
continuous wavelet transform (CWT)

$$C(\text{scale}, \text{position}) = \int_{-\infty}^{\infty} f(t)\psi(\text{scale}, \text{position}, t)dt$$

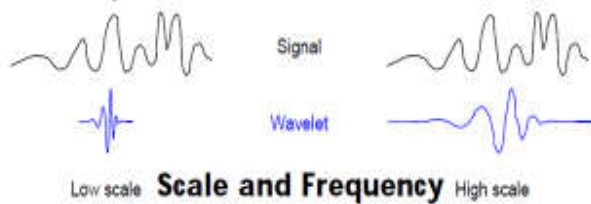
Wavelet Analysis



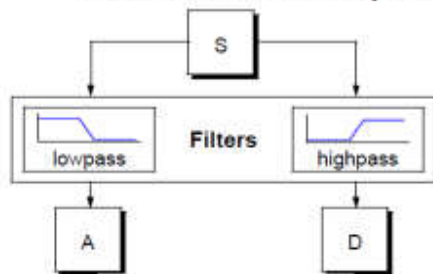
Shifting



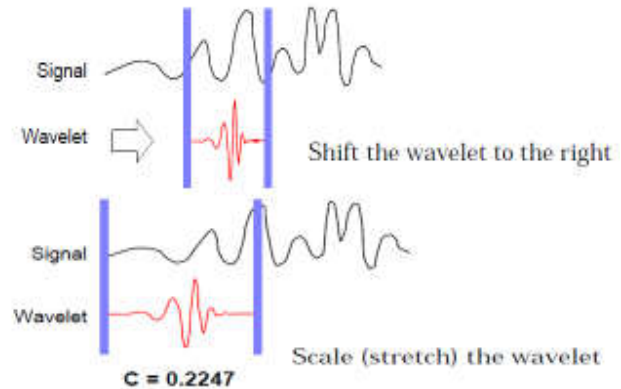
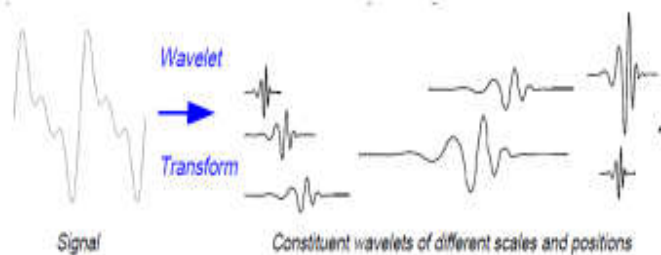
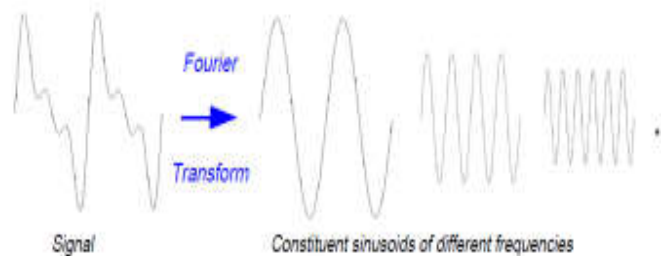
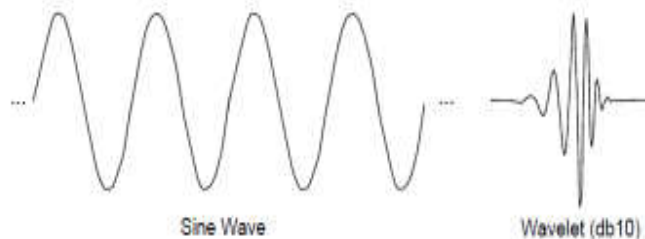
Multiple-Level Decomposition Wavelet Reconstruction



Scale and Frequency

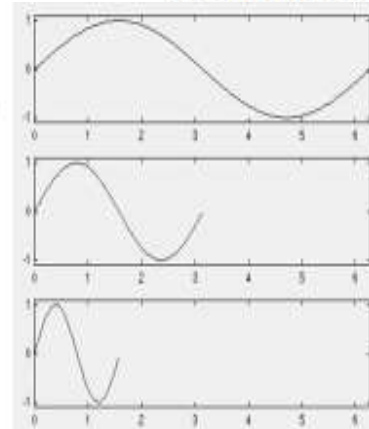


The filtering process

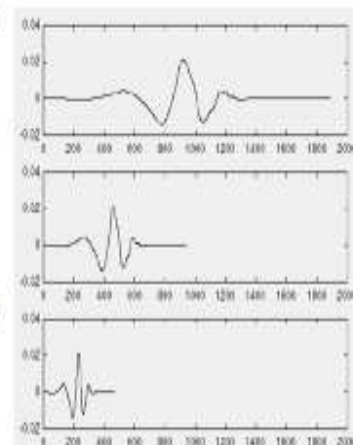


- Low scale $a \Rightarrow$ Compressed wavelet $\Rightarrow$ Rapidly changing details $\Rightarrow$ High frequency ω .
- High scale $a \Rightarrow$ Stretched wavelet $\Rightarrow$ Slowly changing, coarse features $\Rightarrow$ Low frequency ω .

Reconstruction Filters



Scaling



Fourier series

$$F[k] = \int f(t) e^{-j2\pi kt/N} dt \quad f(t) = \frac{1}{2\pi} \sum_{k=0}^{N-1} F[k] e^{j2\pi kt/N}$$

Continuous Fourier transform

$$F(\omega) = \int f(t) e^{-j\omega t} dt \quad f(t) = \frac{1}{2\pi} \int F(\omega) e^{j\omega t} d\omega$$

Discrete Fourier transform

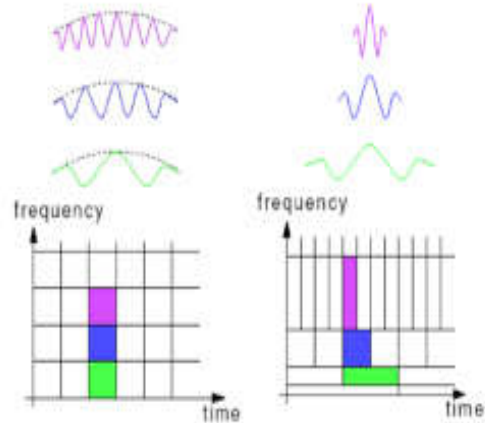
$$F[k] = \sum_{n=0}^{N-1} f[n] e^{-j2\pi nk/N} \quad f[n] = \frac{1}{2\pi} \sum_{k=0}^{N-1} F[k] e^{j2\pi nk/N}$$

Laplace transform

$$F(s) = \int f(t) e^{-st} dt \quad f(t) = \frac{1}{2\pi} \int F(s) e^{st} ds$$

Z-transform

$$F[z] = \sum_{n=0}^{N-1} f[n] e^{-zn} \quad f[n] = \frac{1}{2\pi} \sum_{z=0}^{N-1} F[z] e^{zn}$$



short-time Fourier transform

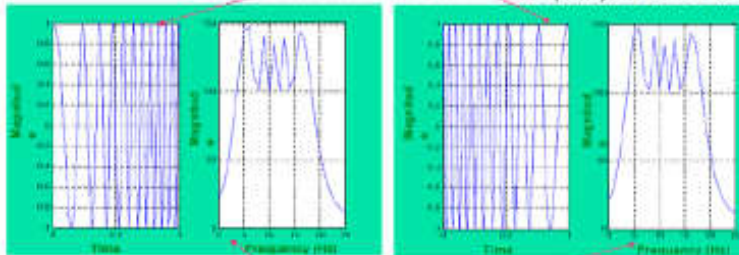
wavelet transform

$$STFT_x^{(w)}(t', f) = \int_t [x(t) \cdot w^*(t - t')] e^{-j2\pi ft} dt$$

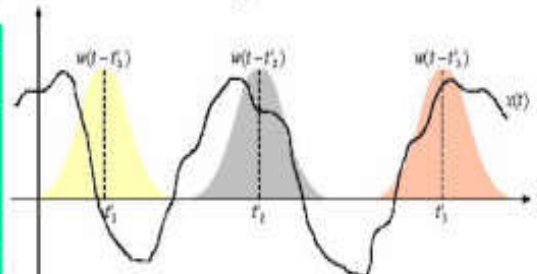
Frequency: 2 Hz to 20 Hz

Different in Time Domain

Frequency: 20 Hz to 2 Hz



Same in Frequency Domain



اعمال پنجره w(t) به سیگنال x(t)

$$e^{j\omega t} = \cos(\omega t) + j \sin(\omega t)$$

$$WPECR = \sum_{i=1}^m \left| \frac{E_{f_i}^{DMG} - E_{f_i}^{UDG}}{E_{f_i}^{UDG}} \right|$$

Damage indicators

Wavelet Packet Energy Change Rates

$$F(\omega) = \int f(t) e^{-j\omega t} dt \quad f(t) = \frac{1}{2\pi} \int F(\omega) e^{j\omega t} d\omega$$

Stationary signals consist of spectral components that do not change in time

- all spectral components exist at all times
- no need to know any time information
- FT works well for stationary signals

تبدیل موجک پیوسته (CWT)

$$CWT_x^\psi(\tau, s) = \Psi_x^\psi(\tau, s) = \frac{1}{\sqrt{|s|}} \int_t x(t) \psi^* \left(\frac{t - \tau}{s} \right) dt$$

Continuous wavelet transform of the signal $x(t)$ using the analysis wavelet $\psi(\cdot)$

The mother wavelet. All kernels are obtained by translating (shifting) and/or scaling the mother wavelet

Scale = 1/frequency

Time resolution: How well two spikes in time can be separated from each other in transform domain

Frequency resolution: How well two spectral components can be separated from each other in the transform domain

$$STFT_x^\omega(t', \omega) = \int_t [x(t) \cdot W(t - t')] \cdot e^{-j\omega t} dt$$

$$X(f) = \int_{-\infty}^{\infty} x(t) \cdot e^{-j2\pi ft} dt$$

$$x(t) = \int_{-\infty}^{\infty} X(f) \cdot e^{j2\pi ft} df$$

رزولوشن زمانی خوب، رزولوشن فرکانسی یائین
رزولوشن فرکانسی خوب، رزولوشن زمانی یائین

پنجره باریک
پنجره پهن

دسته بندی روشهای تحلیل موجک در ارتعاشات

آنالیز موجک پیوسته

آنالیز موجک گسسته

آنالیز بسته موجک

آنالیز موجک نسل دوم (Lifting Scheme)

دسته بندی کاربردهای تحلیل موجک در ارتعاشات

استخراج مشخصه های سیگنال (عیب یابی)

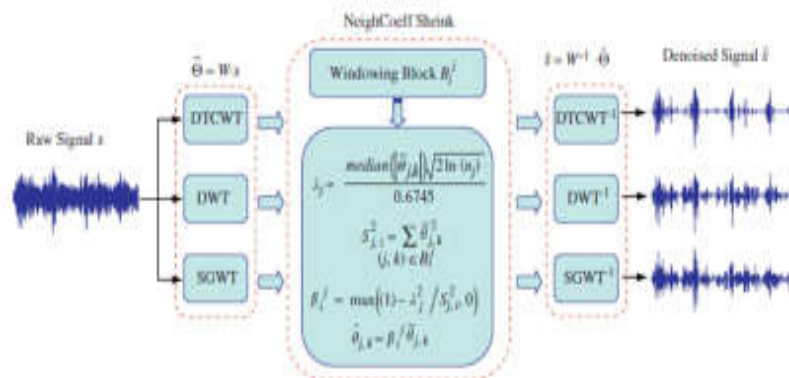
تشخیص نقاط منفرد (عیب یابی)

نویز زدایی و بهبود سیگنال

فشرده سازی سیگنال ارتعاشات

شناسایی سیستم و پارامترهای مودال آن

سایر کاربردها



تابع خودهمبستگی (Autocorrelation)

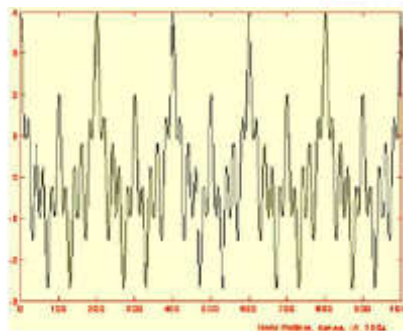
شاخصهای آماری مانند کورتوسیس و ...

آنتروپی شانون

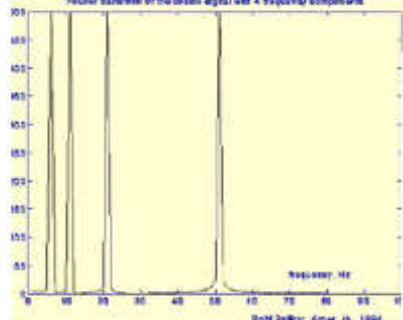
Singular Value Decomposition

دمودولاسیون

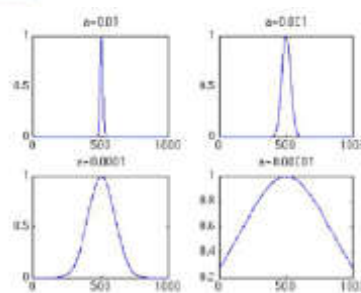
روشهای مختلف بهینه سازی



سیگنال ایستای دارای چهار جزء، فرکانسی ۵، ۲۰، ۱۰ و ۵۰ هرتز

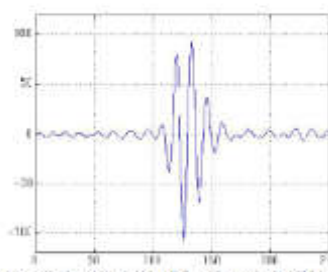


تبدیل فوریه سیگنال

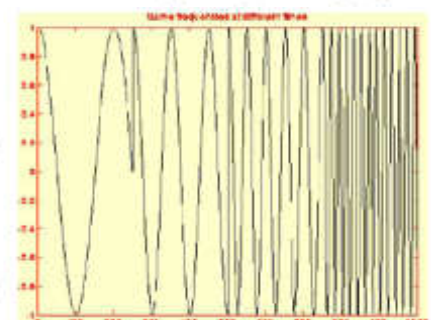


پنجره گوسی با چهار عرض مختلف، عرض پنجره تابعی از پارامتر a است.

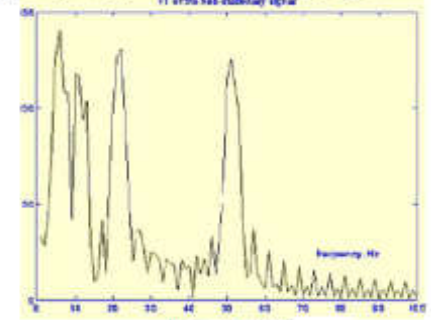
$$w(r) = e^{-\frac{a^2 r^2}{2}}$$



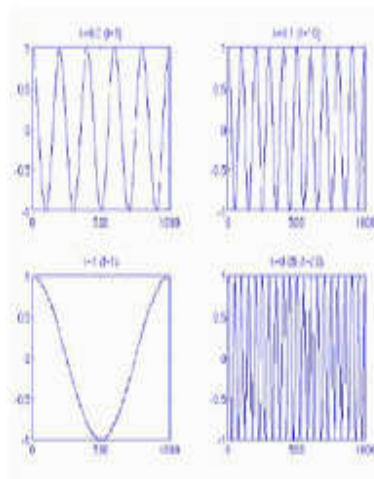
سیگنالی با فرکانسهای پایین در کل سیگنال و فرکانسها و دامنه های بالا در بخش میانی



سیگنال غیرایستای دارای چهار جزء، فرکانسی ۵، ۲۰، ۱۰ و ۵۰ هرتز

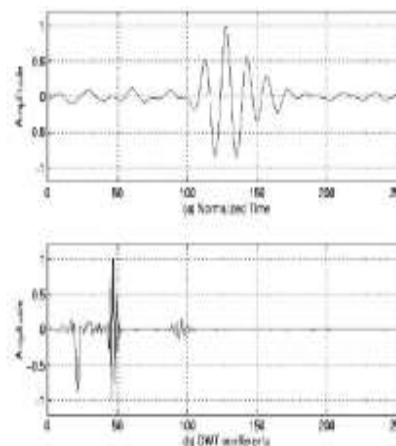


تبدیل فوریه سیگنال



مقیاسهای مختلف یک تابع کیپوس

Synthesis and analysis tools
Wavelet and wavelet packets approaches
Signal and image processing
Discrete and continuous analyses
Orthogonal and redundant approaches
Coding, de-noising and compression approaches



سیگنالهای DWT و فرایند کاهش داده

Wavelet families vary in terms of several important properties. Examples include:

$$x(t) = \frac{1}{C_\psi} \iint \psi_s^*(\tau, s) \frac{1}{s} \psi\left(\frac{t-\tau}{s}\right) d\tau ds \quad \text{تبدیل موجک مملکوس}$$

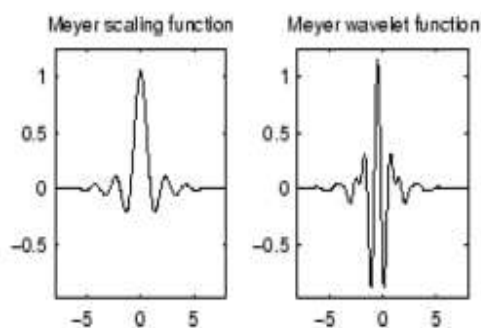
$$C_\psi = \left\{ 2\pi \int_{-\infty}^{\infty} \frac{|\hat{\psi}(\xi)|^2}{|\xi|^2} d\xi \right\}^{1/2} < \infty \quad \text{شرط مقبولیت}$$

گسسته کردن تبدیل موجک پیوسته: سریهای موجک

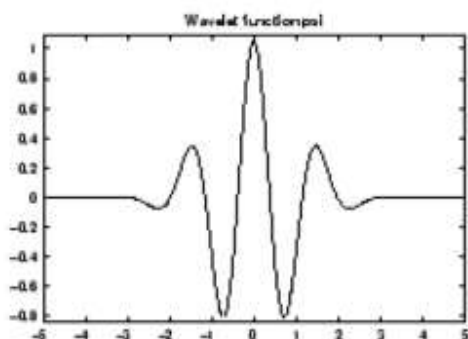
$$N_1 = \frac{x_1}{x_2} N_1 \quad N_2 = \frac{f_2}{f_1} N_1$$

$$N_1 < N_2, \langle f_1 \rangle > \langle f_2 \rangle \text{ یا } S_1 < S_2$$

- Support of the wavelet in time and frequency and rate of decay.
- Symmetry or antisymmetry of the wavelet. The accompanying perfect reconstruction filters have linear phase.
- Number of vanishing moments. Wavelets with increasing numbers of vanishing moments result in sparse representations for a large class of signals and images.
- Regularity of the wavelet. Smoother wavelets provide sharper frequency resolution. Additionally, iterative algorithms for wavelet construction converge faster.
- Existence of a scaling function, ϕ .



Meyer Wavelet



Gaussian Derivative Wavelet gaus8

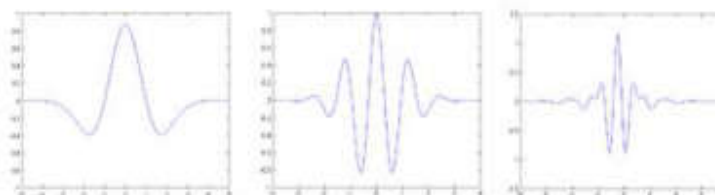
- Wavelet function

$$\psi(\omega) = (2\pi)^{-1/2} e^{i\omega/2} \sin\left(\frac{\pi}{2} v\left(\frac{3}{2\pi}|\omega| - 1\right)\right) \quad \text{if } \frac{2\pi}{3} \leq |\omega| \leq \frac{4\pi}{3}$$

$$\psi(\omega) = (2\pi)^{-1/2} e^{i\omega/2} \cos\left(\frac{\pi}{2} v\left(\frac{3}{4\pi}|\omega| - 1\right)\right) \quad \text{if } \frac{4\pi}{3} \leq |\omega| \leq \frac{8\pi}{3}$$

$$\text{and } \psi(\omega) = 0 \quad \text{if } |\omega| \in \left[\frac{2\pi}{3}, \frac{8\pi}{3}\right]$$

$$\text{where } v(a) = a^4 (35 - 84a + 70a^2 - 20a^3) \quad a \in [0, 1]$$



کلاه مکرکی

مورله

گوس

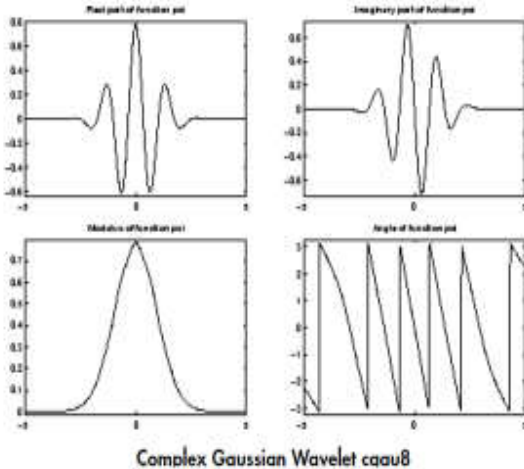
$$w(t) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{t^2}{2\sigma^2}}$$

$$\text{Scaling function } w(t) = \cos(5t) e^{(-t^2/2)}$$

$$\hat{\phi}(\omega) = (2\pi)^{-1/2} \quad \text{if } |\omega| \leq \frac{2\pi}{3}$$

$$\hat{\phi}(\omega) = (2\pi)^{-1/2} \cos\left(\frac{\pi}{2} v\left(\frac{3}{2\pi}|\omega| - 1\right)\right) \quad \text{if } \frac{2\pi}{3} \leq |\omega| \leq \frac{4\pi}{3}$$

$$\hat{\phi}(\omega) = 0 \quad \text{if } |\omega| > \frac{4\pi}{3}$$



■ تبدیل موجک پیوسته (CWT) Continuous wavelet transform

■ تبدیل موجک گسسته (DWT) Discrete wavelet transform

■ تبدیل سریع موجک (FWT) Fast wavelet transform

■ Lifting scheme

■ تجزیه بسته‌های موجک (WPD) Wavelet packet decomposition

■ تبدیل موجک ساکن (SWT) Stationary wavelet transform

موجک‌ها بر مبنای دو عمل اصلی قرار دارند

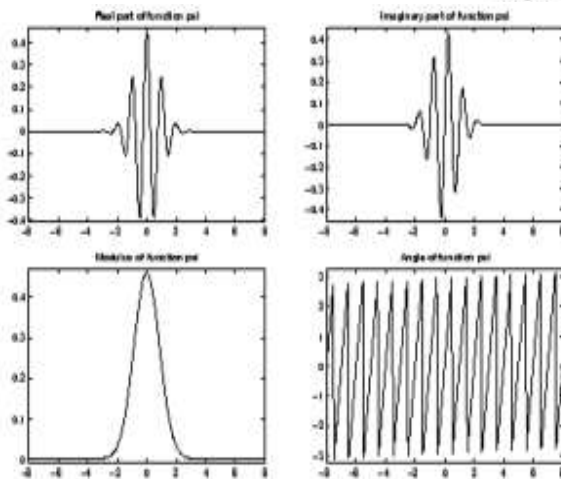
■ انتقال (Translation)

■ تأخیر (Dilation)

مقایسه با تبدیل فوریه

در مقایسه با تبدیل فوریه می‌توان گفت که تبدیل موجک دارای خصوصیت محلی‌سازی بسیار خوبی است. بطور مثال تبدیل فوریه یک پیک تیز دارای تعداد زیادی ضریب است، چرا که توابع پایه تبدیل فوریه توابع سینوسی و کسینوسی هستند که دامنه آنها در کل بازه ثابت است. در حالی که توابع موجک توابعی هستند که بیشتر انرژی آنها در بازه کوچکی متمرکز شده‌است و به سرعت میرا می‌شوند بنابراین با انتخاب مناسب موجک‌های مادر می‌توان فشردگی‌سازی بهتری در مقایسه با تبدیل فوریه انجام داد

در این تبدیل اطلاعات فرکانس و زمان به طور همزمان وجود ندارد. در تبدیل فوریه، اطلاعات مکانی / زمانی در فضای فرکانس از بین می‌رود با انتقال داده‌ها به فضای فرکانس، تمامی اطلاعات مکانی از بین می‌رود. استفاده از تبدیل فوریه باید شرط ثابت بودن سیگنال برقرار باشد تبدیل فوریه تعیین می‌کند که یک فرکانس خاص در سیگنال وجود دارد یا نه و در مورد اینکه این فرکانس در کجای سیگنال واقع شده، اطلاعاتی بدست نمی‌دهد

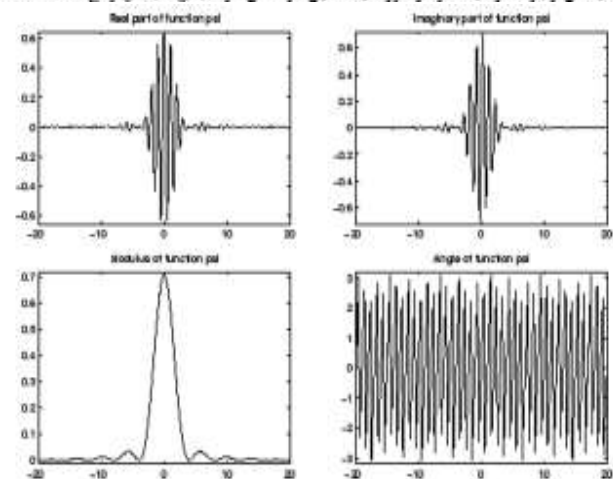


A complex Morlet wavelet is defined by

$$\psi(x) = \frac{1}{\sqrt{\pi f_b}} e^{2i\pi f_c x} e^{-\frac{x^2}{f_b}}$$

depending on two parameters:

- f_b is a bandwidth parameter.
- f_c is a wavelet center frequency.



A complex frequency B-spline wavelet is defined by

$$\psi(x) = \sqrt{f_b} \left(\text{sinc} \left(\frac{f_b x}{m} \right) \right)^m e^{2i\pi f_c x}$$

depending on three parameters:

- m is an integer order parameter ($m \geq 1$).
- f_b is a bandwidth parameter.
- f_c is a wavelet center frequency.

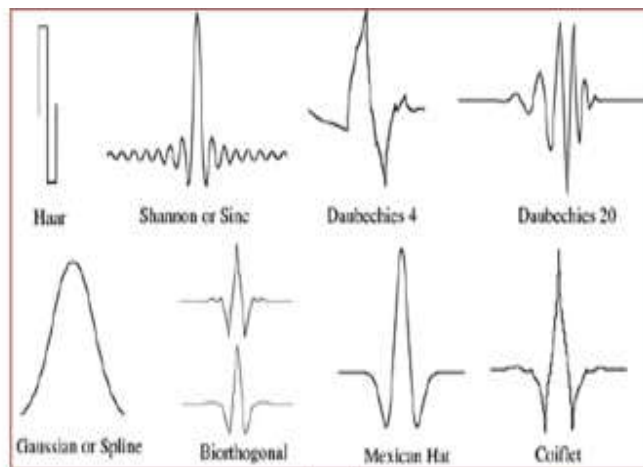
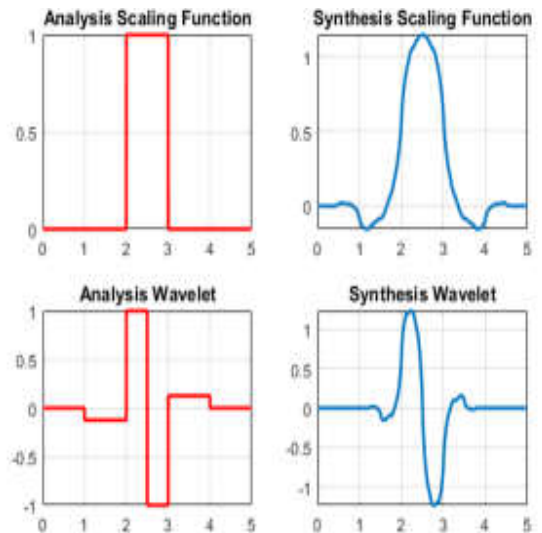
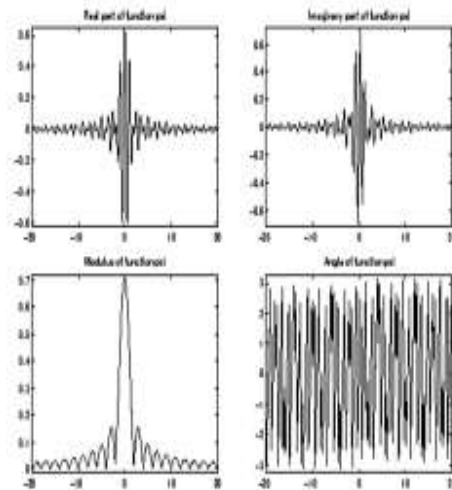


Figure 8
Examples of types of wavelets



The design of the filters is particularly important for the transform to occur correctly and the necessary characteristics are:

- The low-pass filters in the two trees must differ by half a sample period
- Reconstruction filters are the reverse of analysis
- All filters from the same orthonormal set
- Tree a filters are the reverse of tree b filters
- Both trees have the same frequency response



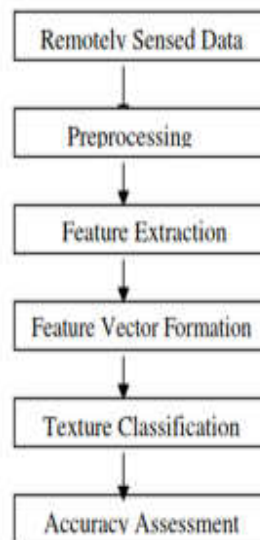
Complex Shannon Wavelet shan 0.5-1

A complex Shannon wavelet is defined by

$$\psi(x) = \sqrt{f_b} \operatorname{sinc}(f_b x) e^{2\pi i f_c x}$$

depending on two parameters:

- f_b is a bandwidth parameter.
- f_c is a wavelet center frequency.



Methodology for texture classification

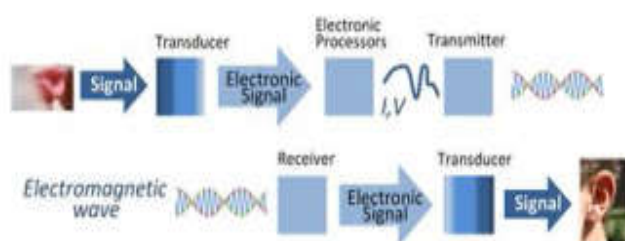
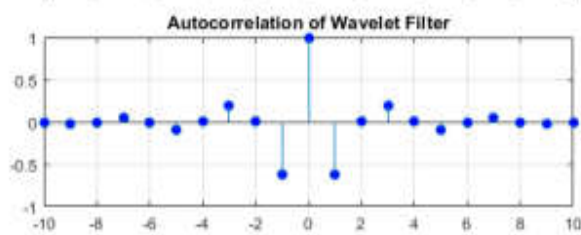
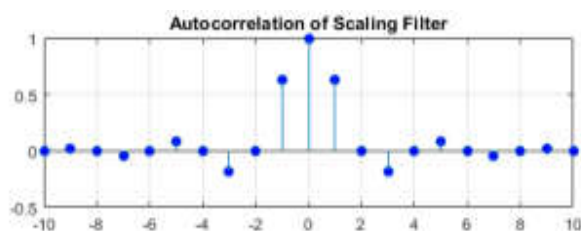
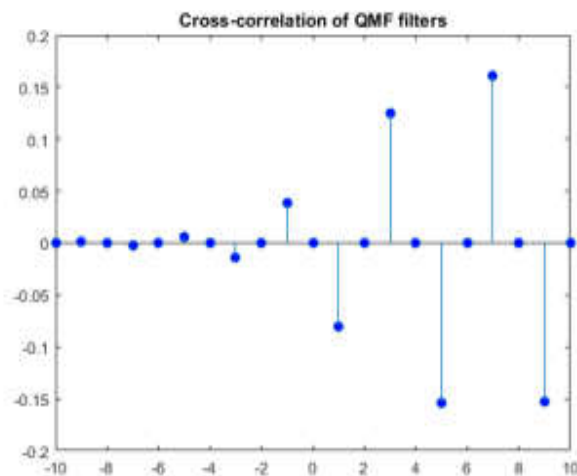
When Δt is large,

1. Bad time resolution
2. Good frequency resolution
3. Low frequency, large scaling factor

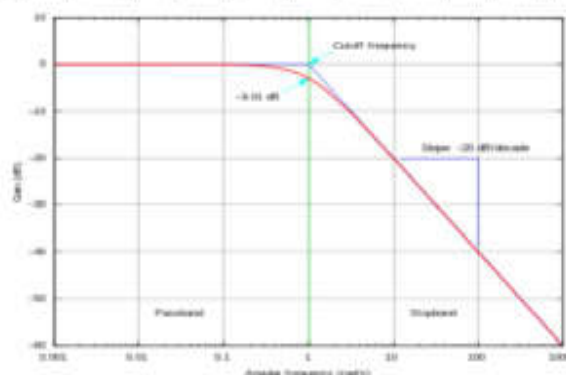
Haar	haar
Daubechies	db
Symlets	sym
Coiflets	coif
BiorSplines	bior
ReverseBior	rbio
Meyer	meyr
DMeyer	dmey
Gaussian	gaus
Mexican_hat	mexh
Morlet	morl
Complex Gaussian	cgau
Shannon	shan
Frequency B-Spline	fbsp
Complex Morlet	cmor
Binlets	binl
Lemarie	lem
Fejer-Korovkin	fk
beylkin	beyl

When Δt is small

1. Good time resolution
2. Bad frequency resolution
3. High frequency, small scaling factor



انتقال سیگنال‌های الکتریکی با استفاده از بردار سیگنال. مدل‌ها امواج فیزیکی را به امواج از نوع جریان الکتریکی یا ولتاژ تبدیل میکنند، که پس از آن بردار می‌شوند، و به شکل امواج الکترومغناطیسی دریافت شده و توسط مدل دیگری به شکل نهایی خود تبدیل می‌شوند.



نمودار دامنه پاسخ فرکانسی یک فیلتر پایین‌گذر

تبدیل هیلبرت توابع منتخب

نماد	شماره	K	t_2	t_1	K^{-1}	u_1	u_2
تبدیل فوری	$\mathcal{F}$	$\frac{e^{-i\omega t}}{\sqrt{2\pi}}$	$-\infty$	∞	$\frac{e^{+i\omega t}}{\sqrt{2\pi}}$	$-\infty$	∞
تبدیل سینوسی فوری	$\mathcal{F}_s$	$\sqrt{\frac{2}{\pi}} \sin(\omega t)$	0	∞	$\sqrt{\frac{2}{\pi}} \sin(\omega t)$	0	∞
تبدیل کسینوسی فوری	$\mathcal{F}_c$	$\sqrt{\frac{2}{\pi}} \cos(\omega t)$	0	∞	$\sqrt{\frac{2}{\pi}} \cos(\omega t)$	0	∞
تبدیل هارتم	$\mathcal{H}$	$\frac{\cos(\omega t) + \sin(\omega t)}{\sqrt{2\pi}}$	$-\infty$	∞	$\frac{\cos(\omega t) + \sin(\omega t)}{\sqrt{2\pi}}$	$-\infty$	∞
تبدیل ملین	$\mathcal{M}$	e^{-t}	0	∞	$\frac{1-e^{-t}}{2\pi i}$	$c-i\infty$	$c+i\infty$
تبدیل لاپلاس دوسویه	$\mathcal{B}$	e^{-st}	$-\infty$	∞	$\frac{e^{+st}}{2\pi i}$	$c-i\infty$	$c+i\infty$
تبدیل لاپلاس	$\mathcal{L}$	e^{-st}	0	∞	$\frac{e^{+st}}{2\pi i}$	$c-i\infty$	$c+i\infty$
تبدیل وایرشتراس	$\mathcal{W}$	$\frac{e^{-(\omega-i\eta)^2/4}}{\sqrt{4\pi}}$	$-\infty$	∞	$\frac{e^{+(\omega-i\eta)^2/4}}{i\sqrt{4\pi}}$	$c-i\infty$	$c+i\infty$
تبدیل هنگل		$i J_0(\omega t)$	0	∞	$v J_0(\omega t)$	0	∞
تبدیل ایل		$\frac{2t}{\sqrt{t^2 - u^2}}$	u	∞	$\frac{-1}{\pi} \frac{d}{\sqrt{u^2 - t^2}}$	t	∞
تبدیل هیلبرت	$\mathcal{H}_H$	$\frac{1}{\pi} \frac{1}{u-t}$	$-\infty$	∞	$\frac{1}{\pi} \frac{1}{u-t}$	$-\infty$	∞
کرل بواسون		$\frac{1-r^2}{1-2r \cos \theta + r^2}$	0	2π			
تبدیل هارم		$\delta(u-t)$	u	$t_2 > u$	$\delta(t-u)$	$u_1 < t$	$u_2 > t$

تبدیل ایل

تبدیل انگرلی

تبدیل رادن

تبدیل فوری

تبدیل فوری کسری

تبدیل لاپلاس

تبدیل لاپلاس دوسویه

تبدیل لاپلاس معکوس

تبدیل ملین

تبدیل موجک پیوسته

تبدیل وایرشتراس

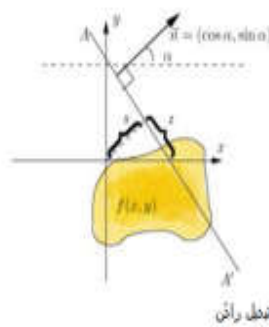
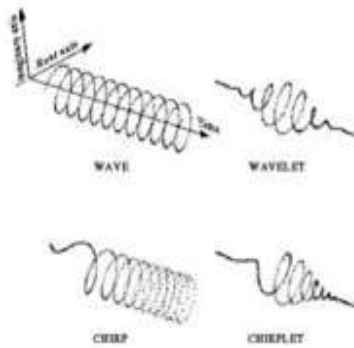
تبدیل هارتم

تبدیل هنگل

تبدیل هیلبرت

تبدیل‌های سینوسی و کسینوسی

تبدیل هیلبرت	تبدیل
$\mathcal{H}(u)(t)$	$u(t)$
$-\cos(t)$	$\sin(t)$
$\sin(t)$	$\cos(t)$
t	1
$t^2 + 1$	$t^2 + 1$
$\frac{1 - \cos(t)}{t}$	$\frac{\sin(t)}{t}$
$\frac{1}{\pi} \ln \left \frac{t + \frac{1}{t}}{t - \frac{1}{t}} \right $	$\frac{1}{\pi} \ln \left \frac{t + \frac{1}{t}}{t - \frac{1}{t}} \right $
$\frac{1}{\pi t}$	$\frac{1}{\pi t}$
$\frac{1}{\pi} \log \left \frac{z-a}{z-b} \right $	$\frac{1}{\pi} \log \left \frac{z-a}{z-b} \right $



تبدیل رادون (به انگلیسی: Radon transform) تبدیلی انتگرالی است که مقدار آن برابر با انتگرال تابع بر روی یک خط است.

$$(s(t), y(t)) = t(\sin \alpha, -\cos \alpha) + s(\cos \alpha, \sin \alpha)$$

$$\mathcal{R}[f](\alpha, s) = \int_{-\infty}^{\infty} f(s(t), y(t)) dt$$

Fractional Fourier transform

$$\mathcal{F}_\alpha[f](u) = \sqrt{1 - i \cot(\alpha)} e^{i\pi \cot(\alpha) u^2} \int_{-\infty}^{\infty} e^{-i2u \cot(\alpha) x - \frac{\cot(\alpha)}{2} x^2} f(x) dx$$

تبدیل فوریه کسری

$$\{\mathcal{M}f\}(s) = \varphi(s) = \int_0^\infty s^{s-1} f(x) dx.$$

تبدیل ملین

به صورت فرم ضربی تبدیل لاپلاس دوسویه در نظر گرفته می‌شود.

$$\{\mathcal{M}^{-1}\varphi\}(x) = f(x) = \frac{1}{2\pi i} \int_{c-i\infty}^{c+i\infty} x^{-s} \varphi(s) ds.$$

تبدیل معکوس نیز اینگونه تعریف می‌شود:

این تبدیل انتگرالی بسیار به سری دیریکلت وابسته است

تابع هینکل مرتبه ν برای تابع $f(r)$ به صورت زیر تعریف می‌شود:

$$F_\nu(k) = \int_0^\infty f(r) J_\nu(kr) r dr$$

که در آن J_ν تابع بسل نوع اول از مرتبه ν است با شرط $\nu \geq -1/2$

معکوس تبدیل هینکل تابع $F_\nu(k)$ نیز به صورت زیر تعریف می‌شود:

$$f(r) = \int_0^\infty F_\nu(k) J_\nu(kr) k dk$$

تبدیل هارتلی تابع $f(t)$ به صورت زیر تعریف می‌شود:

$$H(\omega) = \{\mathcal{H}f\}(\omega) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(t) \cos(\omega t) dt$$

که در آن ω معمولاً فرکانس زاویه‌ای است

$$\cos(t) = \cos(t) + \sin(t) = \sqrt{2} \sin(t + \pi/4) = \sqrt{2} \cos(t - \pi/4)$$

تبدیل هارتلی معکوس خودش است

کرنل‌های پواسون دو بعدی

روی صفحه مختلط کرنل پواسون به صورت زیر داده شده‌است.

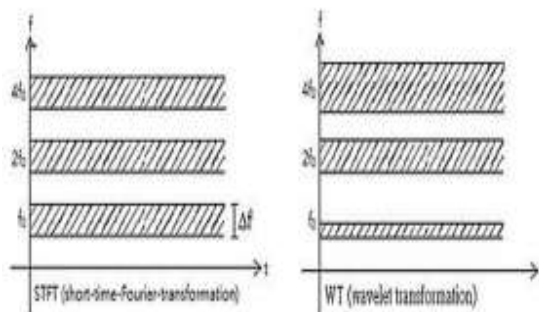
$$P_r(\theta) = \sum_{n=-\infty}^{\infty} r^{|n|} e^{in\theta} = \frac{1-r^2}{1-2r \cos \theta + r^2} = \operatorname{Re} \left(\frac{1+re^{i\theta}}{1-re^{i\theta}} \right), \quad 0 \leq r < 1$$

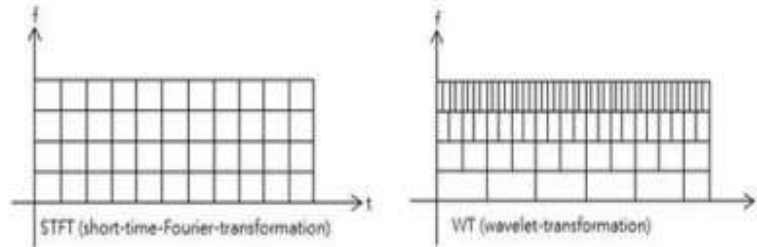
$$P(y) = 2 \int_1^\infty \frac{f(r)r}{\sqrt{r^2-y^2}} dr$$

تبدیل ابل، تبدیلی انتگرالی است که اغلب در تحلیل توابع دارای تقارن کروی یا محوری به کار می‌رود.

$$f(r) = -\frac{1}{\pi} \int_1^\infty \frac{dF}{dy} \frac{1}{\sqrt{y^2-r^2}} dy$$

عکس این تبدیل به صورت زیر به دست می‌آید:





This shows that wavelet transformation is good in time resolution of high frequencies, while for slowly varying functions, the frequency resolution is remarkable.

Fourier Transform

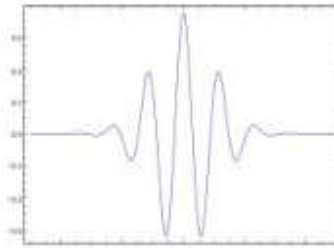
$$\text{Fourier Transform } \hat{f}(\omega) = \int_{-\infty}^{\infty} f(t) e^{-2\pi i \omega t} dt$$

$$\text{Inverse Fourier Transform } f(t) = \int_{-\infty}^{\infty} \hat{f}(\omega) e^{2\pi i \omega t} d\omega$$

$$\text{Parseval Theorem } \int_{-\infty}^{\infty} |f(t)|^2 dt = \int_{-\infty}^{\infty} |\hat{f}(\omega)|^2 d\omega$$

$$\text{Discrete Fourier Transform } \hat{f}_n \equiv \frac{1}{N} \sum_{k=0}^{N-1} f_k e^{-2\pi i k n / N}$$

$$n = -\frac{N}{2} + 1, \dots, \frac{N}{2}$$



Real-valued Morlet wavelet



Complex-valued Morlet wavelet

Low-frequency	Bad clarity in time domain	Good clarity in frequency domain
High-frequency	Bad clarity in frequency domain	Good clarity in time domain

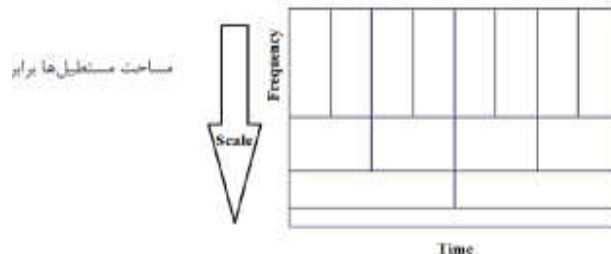
Transform	Representation	Input
Fourier transform	$\hat{f}(\xi) = \int_{-\infty}^{\infty} f(x) e^{-2\pi i x \xi} dx$	ξ , frequency
Time-frequency analysis	$X(t, f)$	t , time; f , frequency
Wavelet transform	$X(a, b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} \psi\left(\frac{t-b}{a}\right) x(t) dt$	a , scaling; b , time

Wavelet compression
data compression image compression
video compression and audio compression
Gaussian

• برای سیگنال‌های گذرا
Hamming and Hanning
• سیگنال‌های تصادفی باند باریک

پنجره‌های یا طول متغیر برای فرکانس‌های مختلف استفاده می‌شوند:

- آنالیز فرکانس‌های بالا ← استفاده از پنجره‌های باریکتر برای رزولوشن زمانی بهتر
 - آنالیز فرکانس‌های پایین ← استفاده از پنجره‌های عریض برای رزولوشن فرکانسی بهتر
- رزولوشن‌های زمانی و فرکانسی در تبدیل موجک

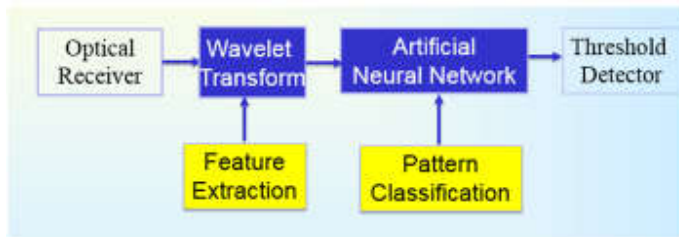


افزایش مقیاس = افزایش بازه زمانی = کاهش رزولوشن زمانی = افزایش رزولوشن فرکانسی

موجک WAVELET

$$wt(s, \tau) = \langle x, \psi_{s, \tau} \rangle = \frac{1}{\sqrt{s}} \int_{-\infty}^{\infty} x(t) \psi^*\left(\frac{t-\tau}{s}\right) dt$$

در رابطه فوق s فاکتور مقیاس است.
مقدار τ بیان کننده میزان شیفت است.
زمان به صورت مستقیم و فرکانس به صورت معکوس با فاکتور مقیاس در ارتباط است.



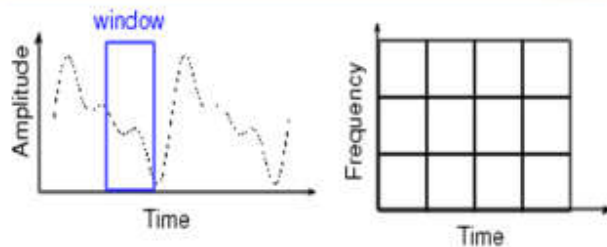
Random signals

Stationary signals:

- ☐ Statistics don't change with time
- ☐ Frequency contents don't change with time
- ☐ Information doesn't change with time

Non-stationary signals:

- ☐ Statistics change with time
- ☐ Frequencies change with time
- ☐ Information quantity increases



$$\text{STFT}_x^{(\omega)}(t', f) = \int_t [x(t) \cdot \omega^*(t - t')] \cdot e^{-j2\pi ft} dt$$

$\omega(t)$: the window function

Dennis Gabor

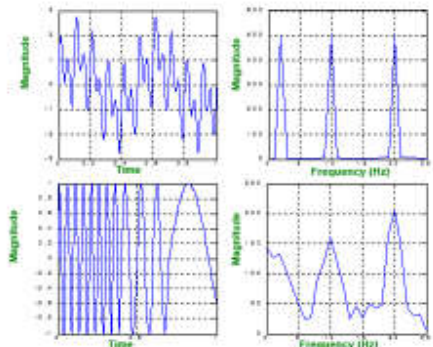
Short-time Fourier Transform

2 Hz + 10 Hz + 20Hz

Stationary

0.0-0.4: 2 Hz +
0.4-0.7: 10 Hz +
0.7-1.0: 20Hz

Non-Stationary



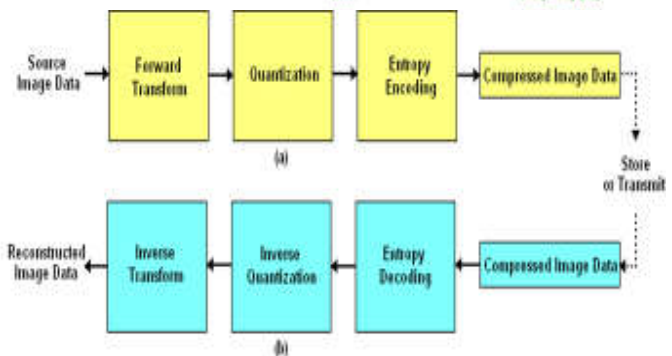
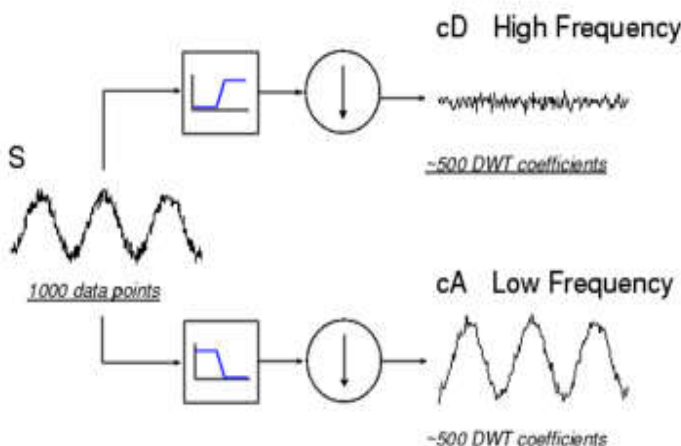
Continuous Wavelet Transform

$$\text{CWT}_x^\psi(\tau, s) = \Psi_x^\psi(\tau, s) = \frac{1}{\sqrt{|s|}} \int x(t) \cdot \psi^*\left(\frac{t - \tau}{s}\right) dt$$

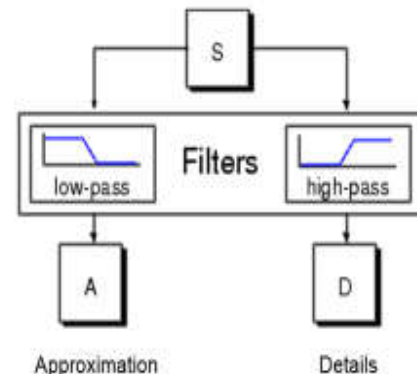
Translation
(The location of the window)

Scale

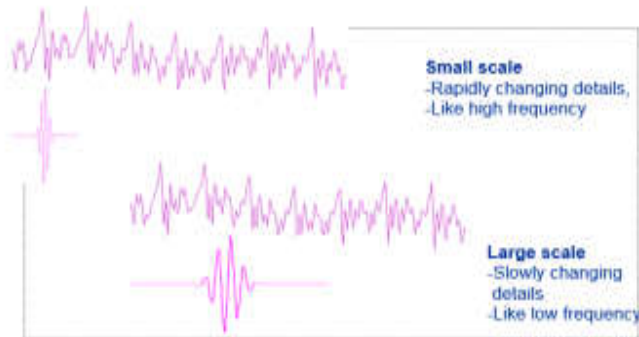
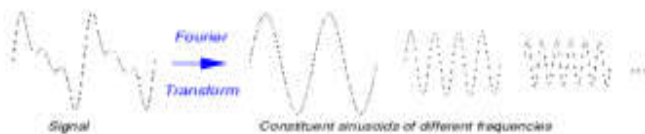
Mother Wavelet



Discrete Wavelet Transform



Fourier transform:
$$F(\omega) = \int_{-\infty}^{\infty} f(t)e^{-j\omega t} dt$$

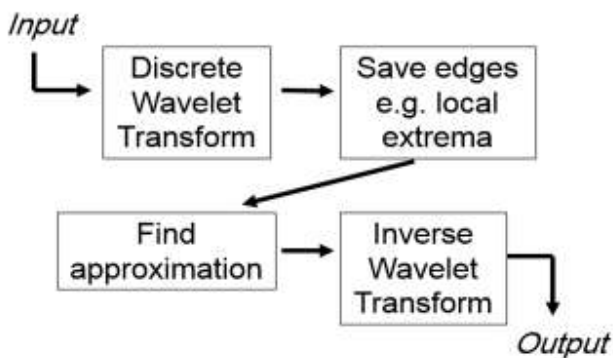
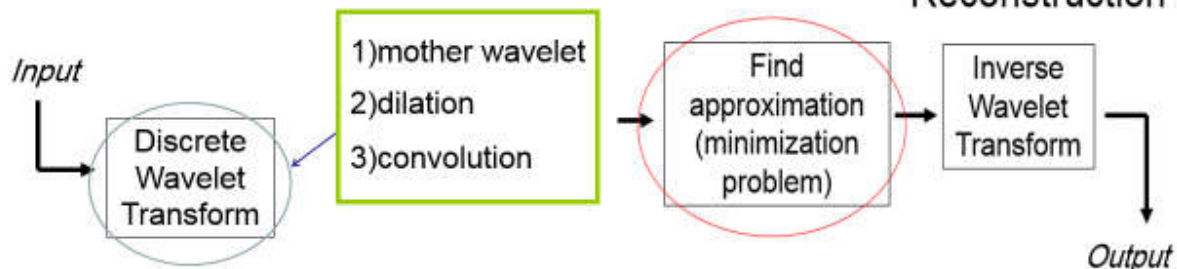


The Continuous Wavelet Transform

$$W_{\psi}(s, \tau) = \int_{-\infty}^{\infty} f(x)\psi_{s, \tau}(x)dx \quad \psi_{s, \tau}(x) = \frac{1}{\sqrt{s}}\psi\left(\frac{x-\tau}{s}\right)$$

$$f(x) = \frac{1}{C_{\psi}} \int_0^{\infty} \int_{-\infty}^{\infty} W_{\psi}(s, \tau) \frac{\psi_{s, \tau}(x)}{s^2} d\tau ds \quad C_{\psi} = \int_{-\infty}^{\infty} \frac{|\Psi(u)|^2}{|u|} du$$

(best match operation)



Wavelet Series Expansion

$$f(x) = \sum_k c_{j_0}(k) \phi_{j_0, k}(x) + \sum_{j=j_0}^{\infty} \sum_k d_j(k) \psi_{j, k}(x)$$

$$c_{j_0}(k) = \int f(x) \phi_{j_0, k}(x) dx$$

$$d_j(k) = \int f(x) \psi_{j, k}(x) dx$$

Discrete Wavelet Transform

$$W_{\phi}(j_0, k) = \frac{1}{\sqrt{M}} \sum_x f(x) \phi_{j_0, k}(x)$$

$$W_{\psi}(j, k) = \frac{1}{\sqrt{M}} \sum_x f(x) \psi_{j, k}(x)$$

$$f(x) = \frac{1}{\sqrt{M}} \sum_k W_{\phi}(j_0, k) \phi_{j_0, k}(x) + \frac{1}{\sqrt{M}} \sum_{j=j_0}^{\infty} \sum_k W_{\psi}(j, k) \psi_{j, k}(x)$$

Fourier Transform

- Continuous Fourier Transform
- Short-Time Fourier Transform (STFT)

Wavelet Transform

- Multi-Resolution Analysis (MRA)
- Continuous Wavelet Transform (CWT)
- Wavelet Functions Properties

Discrete Wavelet Transform

- Discrete and Continuous Wavelet Transform
- Subband Coding Algorithm
- Matrix Interpretation

Wavelet Transform Applications

- Discontinuity Detection in the ECG Signal
- Image Compression
- Image Segmentation
- Noise Reduction by Wavelet Shrinkage

Wavelet Transform

$$W(\lambda, u) = \int_{-\infty}^{\infty} f(t) \psi_{\lambda, u}(t) dt \quad \psi_{\lambda, u}(t) \equiv \frac{1}{\sqrt{\lambda}} \psi\left(\frac{t-u}{\lambda}\right) \quad \lambda > 0$$

Gives good time resolution for high frequency events, and good frequency resolution for low frequency events, which is the type of analysis best suited for many real signals.

Mother wavelet properties

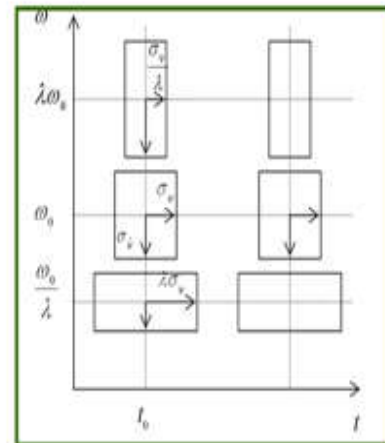
$$\int_{-\infty}^{\infty} \psi(t) dt = 0 \quad \int_{-\infty}^{\infty} |\psi(t)| dt < \infty \quad \int_{-\infty}^{\infty} |\psi(t)|^2 dt = 1$$

$$\omega_{\psi_{\lambda, t}}^0 = \frac{\int_0^{\infty} \omega |\hat{\psi}_{\lambda, t}(\omega)|^2 d\omega}{\int_0^{\infty} |\hat{\psi}_{\lambda, t}(\omega)|^2 d\omega}$$

$$\sigma_{\psi_{\lambda, t}} \quad \sigma_{\dot{\psi}_{\lambda, t}}$$

$$\sigma_{\dot{\psi}_{\lambda, t}} = \frac{\sigma_{\dot{\psi}_{1,0}}}{\lambda} \quad \omega_{\dot{\psi}_{\lambda, t}}^0 = \frac{\omega_{\dot{\psi}_{1,0}}^0}{\lambda}$$

$$\left[t_0 \pm \lambda \sigma_{\psi_{1,0}} \times \frac{\omega_{\dot{\psi}_{1,0}}^0}{\lambda} \pm \frac{\sigma_{\dot{\psi}_{1,0}}}{\lambda} \right]$$



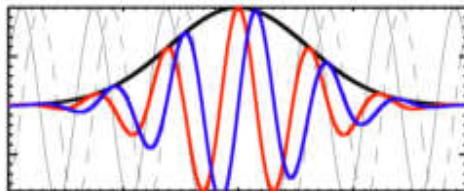
Some Continuous Wavelets

Morlet

$$\pi^{-1/4} e^{-i\omega_0 \eta} e^{-\eta^2/2}$$

$$\psi_{\lambda, u}(t) \equiv \frac{1}{\sqrt{\lambda}} \psi\left(\frac{t-u}{\lambda}\right)$$

Morlet

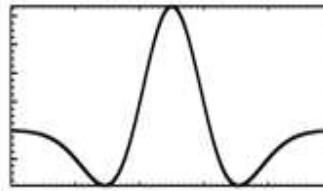


Gabor

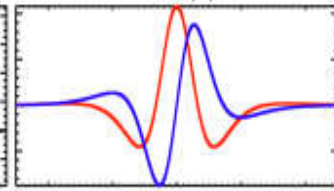
$$g_{\omega, u}(t) \equiv g(t-u) e^{-i2\pi \omega t}$$

$$g(u) = \pi^{-1/4} e^{-u^2/2}$$

Mexican Hat



Paul (4)



Discrete Wavelet Transform

DWT is defined as:

$$f(t) = \sum_k \sum_j a_{jk} \psi_{jk}(t)$$

$$a_{jk} = \int f(t) \psi_{jk}(t) dt$$

$$\psi_{jk}(t) = 2^{j/2} \psi(2^j t - k)$$

Definition – The 'continuous' wavelet transform is defined as follows:

$$\Psi_x^\psi(\tau, s) = \frac{1}{\sqrt{|s|}} \int x(t) \cdot \psi^*\left(\frac{t-\tau}{s}\right) dt$$

For energy normalization

scaling

translation

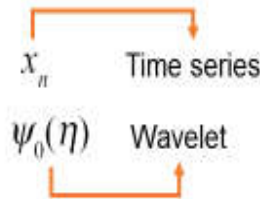
Haar Transform

Haar wavelet function

$$\psi(x) = \begin{cases} 1 & 0 \leq x < \frac{1}{2} \\ -1 & \frac{1}{2} \leq x < 1 \\ 0 & \text{otherwise} \end{cases}$$

$$\psi_{j,k}(x) = \psi(2^j x - k)$$

Continuous Wavelet Transform For Discrete Data



$$W_n(s) = \sum_{n'=0}^{N-1} x_{n'} \psi^* \left[\frac{(n' - n)\delta t}{s} \right]$$

Defined as the convolution with a scaled $\psi_0(\eta)$ and translated version of N times for each s : Slow!

Hat function

$$g = \begin{cases} 0, & x < 0 \\ 1, & x \in [0, 1] \\ 0, & x > 1 \end{cases}$$

Using the convolution theorem, the wavelet transform is the inverse Fourier transform

$$W_n(s) = \sum_{k=0}^{N-1} \hat{x}_k \hat{\psi}^*(s\omega_k) e^{i\omega_k n \delta t}$$

Gauss function

$$g(t) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(t-t_0)^2}{2\sigma^2}\right)$$

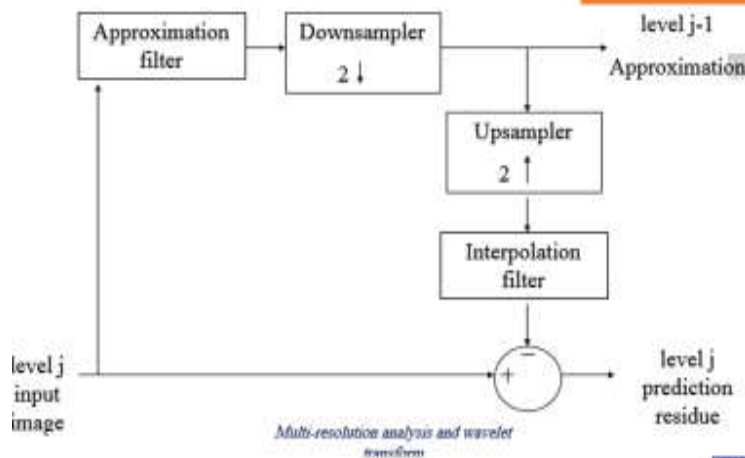
Gabor function

$$g(t) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp(i\Omega(t-t_0) - i\theta) \exp\left(-\frac{(t-t_0)^2}{2\sigma^2}\right)$$

$$\hat{x}_k = \frac{1}{N} \sum_{n=0}^{N-1} x_n e^{-2\pi i k n / N}$$

DFT (FFT) of the time series

Torrence and Compo (1998)



Haar wavelet

$$D(1,0) = \sqrt{2} \int_{-\infty}^{\infty} \psi^{(H)}(u) x(u) du$$

where

$$\psi^{(H)}(u) = \begin{cases} -\frac{1}{\sqrt{2}}, & -1 < u \leq 0 \\ \frac{1}{\sqrt{2}}, & 0 < u \leq 1 \\ 0, & \text{otherwise} \end{cases}$$

Some traditional Multi-Resolution Analysis techniques are:

- Sub-band coding
- Image pyramids
- Short Term Fourier Transform (STFT)

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

Continuous Fourier Transform (FT)

$$X(\omega) = \int_{-\infty}^{+\infty} x(t) e^{-j\omega t} dt \quad \int_{-\infty}^{+\infty} |x(t)|^2 dt < \infty$$

Inverse FT $x(t) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} X(\omega) e^{j\omega t} d\omega$

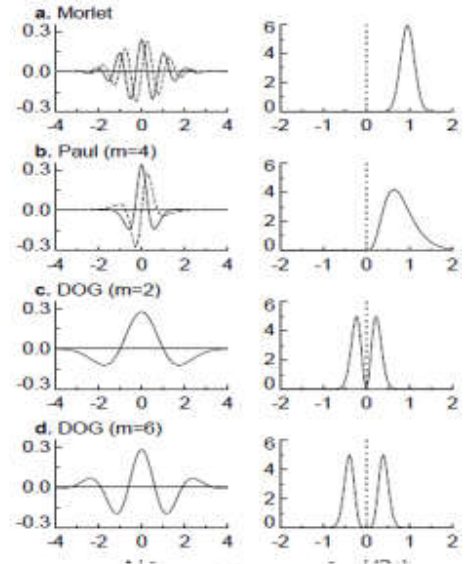
Discrete Short-Time Fourier Transform (STFT)

CWT of the signal $x(t)$:

$$W_x^{\psi}(s, u) = \langle x, \psi_{u,s} \rangle = \frac{1}{\sqrt{s}} \int_{-\infty}^{+\infty} x(t) \psi^* \left(\frac{t-u}{s} \right) dt$$

$$X(u, k) = \sum_{n=0}^{N-1} x(n) w(n-u) e^{-j2\pi kn/N}$$

Name	$\psi_f(t)$	$\hat{\psi}_f(\omega)$	c-folding time ξ	Fourier wavelength λ
Morlet (ω_0 = frequency)	$\pi^{-1/2} e^{i\omega_0 t} e^{- t ^2/2}$	$\pi^{-3/4} H(\omega) e^{-i(m-\omega_0)^2/2}$	$\sqrt{2}\xi$	$\frac{4\pi\xi}{\omega_0 + \sqrt{2 + \omega_0^2}}$
Pauli (m = order)	$\frac{2^m t^m m!}{\sqrt{\pi(2m)!}} (1 - it)^{-(m+1)}$	$\frac{2^m}{\sqrt{\pi(2m-1)!}} H(\omega) (i\omega)^m e^{-\omega^2/2}$	$\xi/\sqrt{2}$	$\frac{4\pi\xi}{2m+1}$
DOG (m = derivative)	$\frac{(-1)^{m+1}}{\sqrt{\Gamma(m+\frac{1}{2})}} \frac{d^m}{dt^m} (e^{-t^2/2})$	$\frac{-i^m}{\sqrt{\Gamma(m+\frac{1}{2})}} (i\omega)^m e^{-\omega^2/2}$	$\sqrt{2}\xi$	$\frac{2\pi\xi}{\sqrt{m+\frac{1}{2}}}$

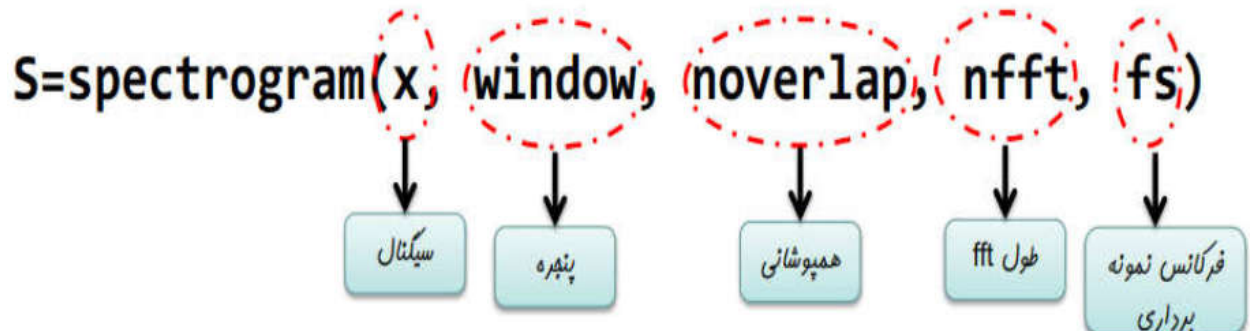


$$f(t) = \int_{R^+} \int_R c(s, \tau) \Psi_{s,\tau}(t) \, ds \, d\tau .$$

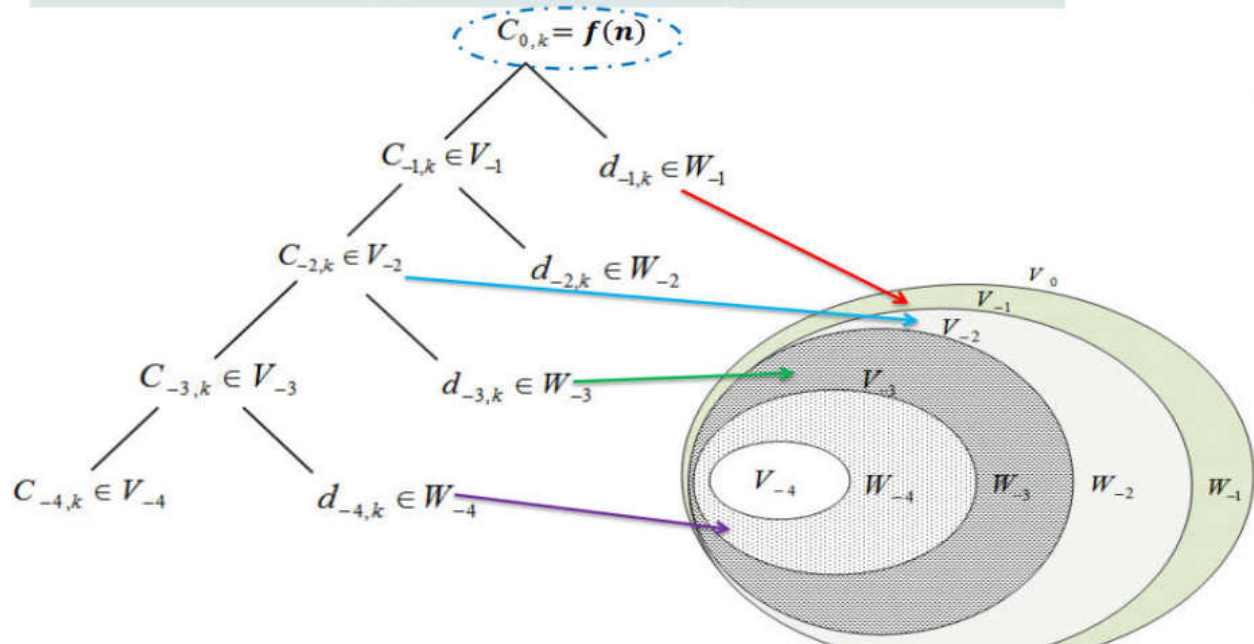
Inverse continuous wavelet transform

- ♦ The mother wavelet should be oscillatory $\int_{-\infty}^{\infty} \Psi(t) \, dt = 0$.
- ♦ Mother wavelet has to have a finite energy $\int_{-\infty}^{\infty} |\Psi(t)|^2 \, dt < \infty$.

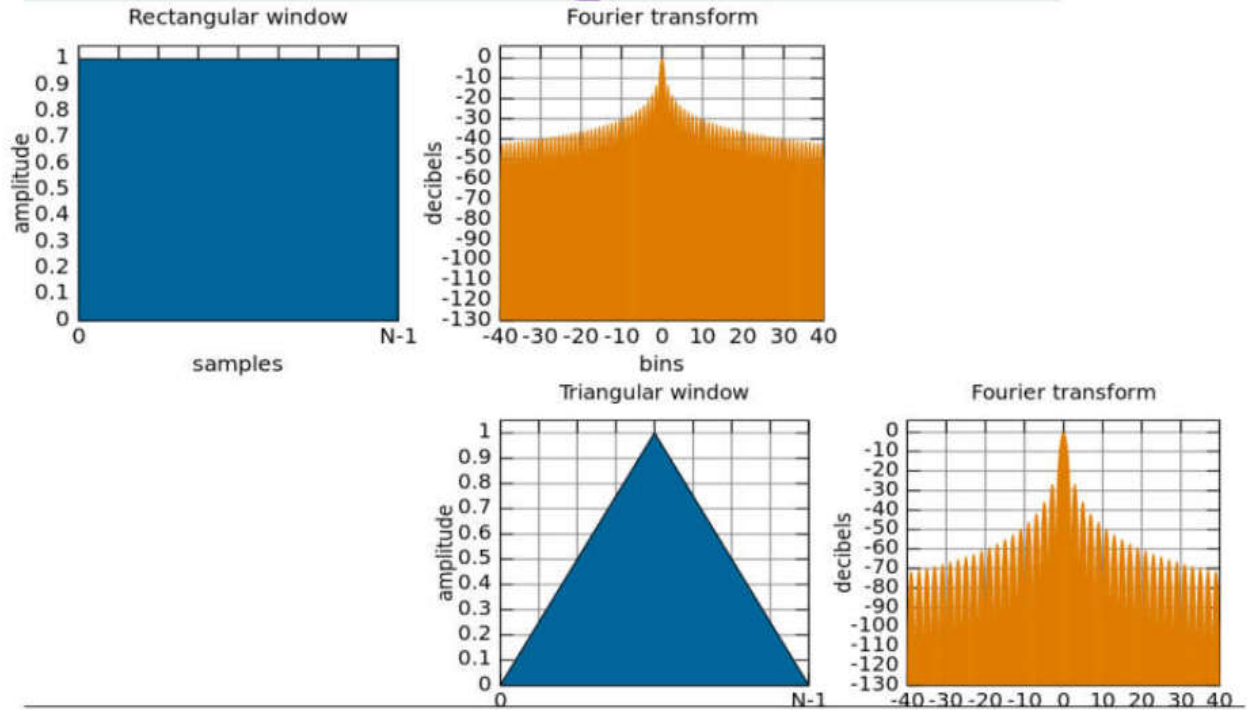
- برنامه نویسی مستقیم بر اساس تبدیل فوریه زمان-کوتاه گسسته
- استفاده از دستور spectrogram
- مثال :



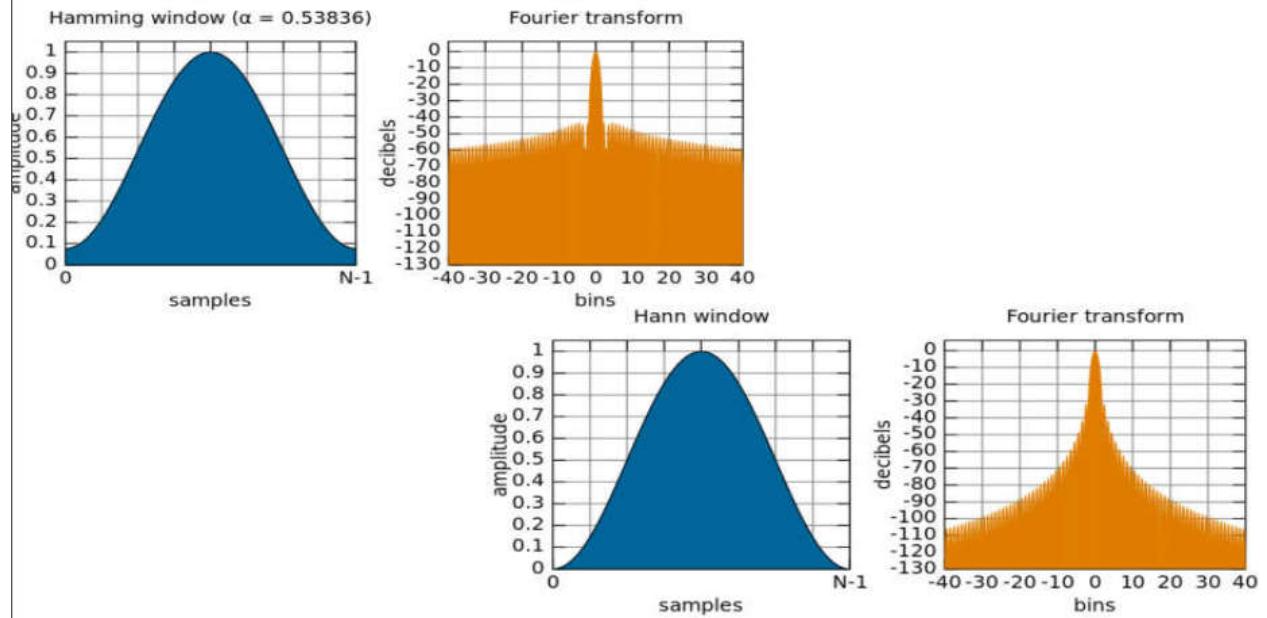
درخت تجزیه موجک



انواع پنجره



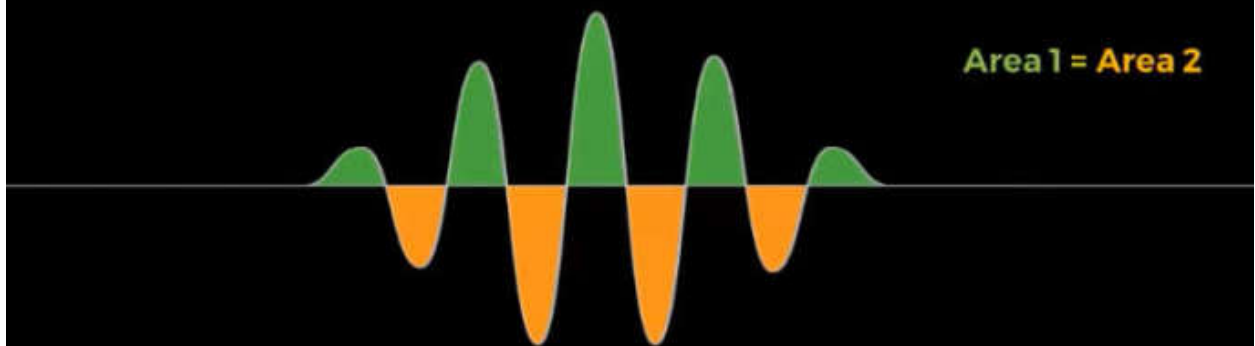
انواع پنجره



FOURIER TRANSFORM



WAVELET TRANSFORM



FOURIER TRANSFORM

$$X(F) = \int_{-\infty}^{\infty} x(t) e^{-j2\pi Ft} dt$$

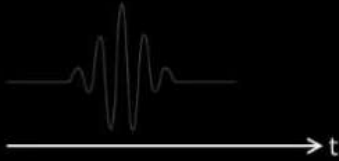
frequency $\swarrow$ $X(F)$ $\nwarrow$ $x(t)$ $\searrow$ time

WAVELET TRANSFORM

$$X(a, b) = \int_{-\infty}^{\infty} x(t) \psi_{a,b}^*(t) dt$$

$\nwarrow$ $x(t)$ $\searrow$ time

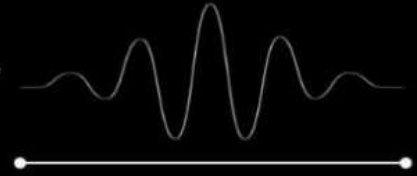
TRANSLATION AND SCALE



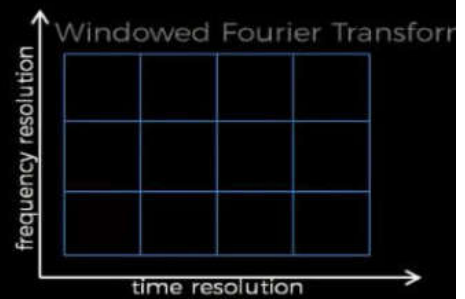
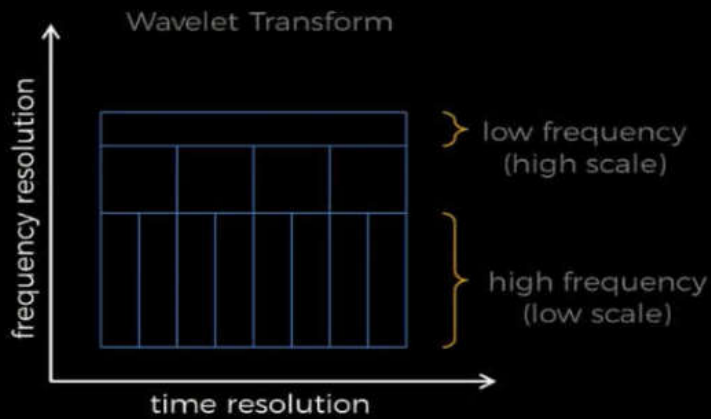
high-frequency
scale
(low scale)



low-frequency
scale
(high scale)

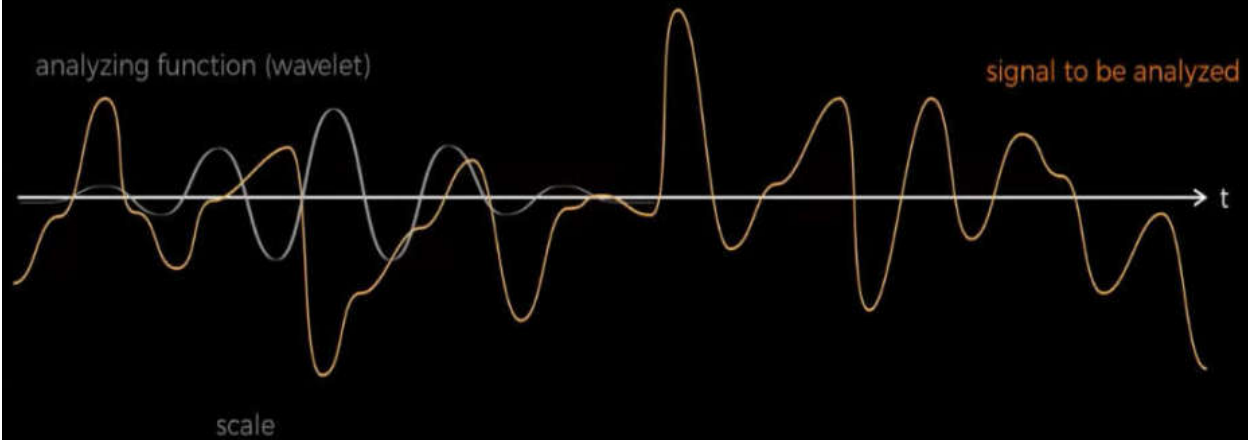


RESOLUTION



CORRELATION

$$X(a, b) = \int_{-\infty}^{\infty} x(t) \psi_{a,b}^*(t) dt$$



Heisenberg uncertainty:

SELECTIVITY IN FREQUENCY

VANISHING MOMENTS

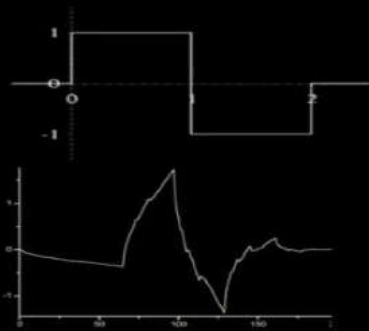
$$m_k = \int_{-\infty}^{\infty} f(x) x^k dx$$

More selective wavelet = less compact support (less selective in time)

Moment k "vanishes" if the integral is zero

REGULARITY

Low Regularity:



Higher Regularity:

