畳み込みとGCN

今回の解説の流れ

畳み込みって何？

- そもそも何をしたい概念？
- 教科書に出てくる数式と、GCNで使う数式が違うように見えるのはなぜ？
- 1次元の例で考える畳み込み（Spatialな場合）
- 1次元の例で考える畳み込み（Spectralな場合）

Spatial GCN

- Spatial GCNの概要
- Spatial GCNの計算コスト

Spectral GCN

- Spectral GCNの概要
- Spectral GCNの計算コスト

畳み込みって何？

そもそも何をしたい概念？

$$y(t) = g(t) * x(t)$$

もとの関数にフィルターをかける！

- 何のために？

→用途は様々！（だから概要をつかみにくい…）

今回は特徴をはっきりさせるため

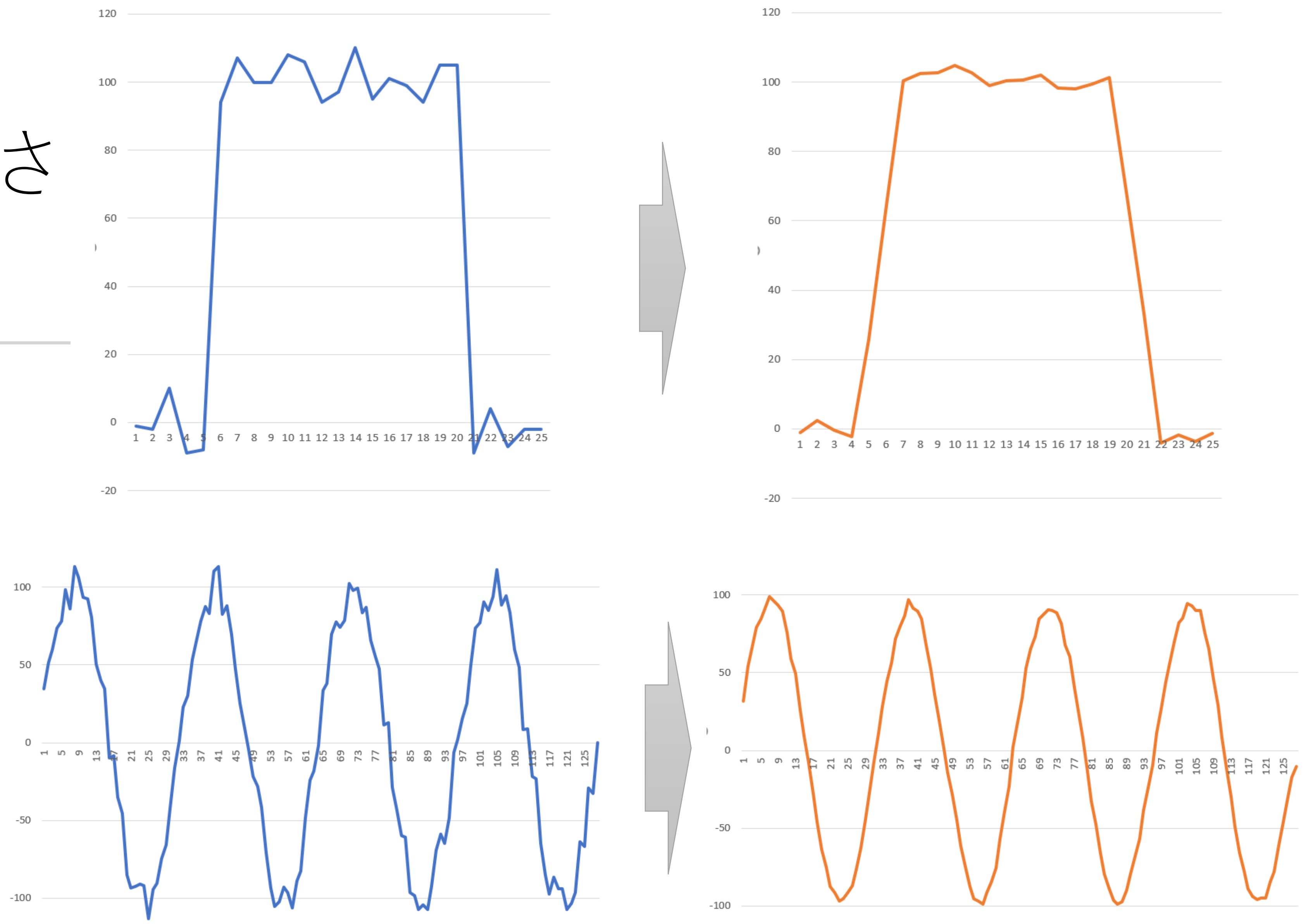
特徴をはっきりさせる？

- これを2次元画像に対して使う

  →CNN
- これをグラフ（ネットワーク）に対して使う

  →GCN

畳み込み前　　畳み込み後



畳み込みで関数の特徴を際立たせている！

教科書の数式とは違うように見えるのは？

私たちは…

$$x_{k+1,j} = L_k h \left(\sum_{i=1}^{f_{k-1}} \boxed{F_{k,i,j}} x_{k,i} \right) \quad (j = 1 \ldots f_k)$$

ここが「重み」（のようなもの）

な式を読めるようになって、活用したい！

しかし、畳み込みについて、書籍を開いても、web上の情報を検索しても

$$(f * g)(t) = \int f(\tau) g(t - \tau)\, d\tau \qquad や \qquad (f * g)(m) = \sum_n f(n)\, g(m - n)$$

$t - \tau$という部分が気になるところだが、より重要なことは『元の関数に「重み」をかけて、強調したり、捨象したりしている』という構造！

という式しか出てこない…なぜ？

「因果的な時不変システム」を取り扱う際に便利な形式だから！

「過去の出来事が、現在にどれだけ影響しているのか」といったことをモデル化する際に便利！（ここでは詳しくは説明しない。）

一般的（より広い意味）に考えると…

→元の関数に「重み」をかけて、強調するところと、捨象するところの区別をする　ということがキモ！

畳み込みの一般的な形

連続的な場合

$$y(t) = (f * g)(t) = \int f(\tau)\, \boxed{g(t,\tau)}\, d\tau$$

重み

離散的な場合

$$y(m) = (f * g)(m) = \sum_{n} f(n)\, \boxed{g(m,n)}$$

具体例　（n は0から4まで変化させる）

n	0	1	2	3	4
f	0	2	4	6	8

$m-n$	-1	0	1
g	5	3	1

m	0	1	2	3	4
y		0×1＋1×3＋4×5＝23	2×1＋4×3＋6×5＝44	4×1＋6×3＋8×5＝62	

$$f(n) = \{0,2,4,6,8\}$$

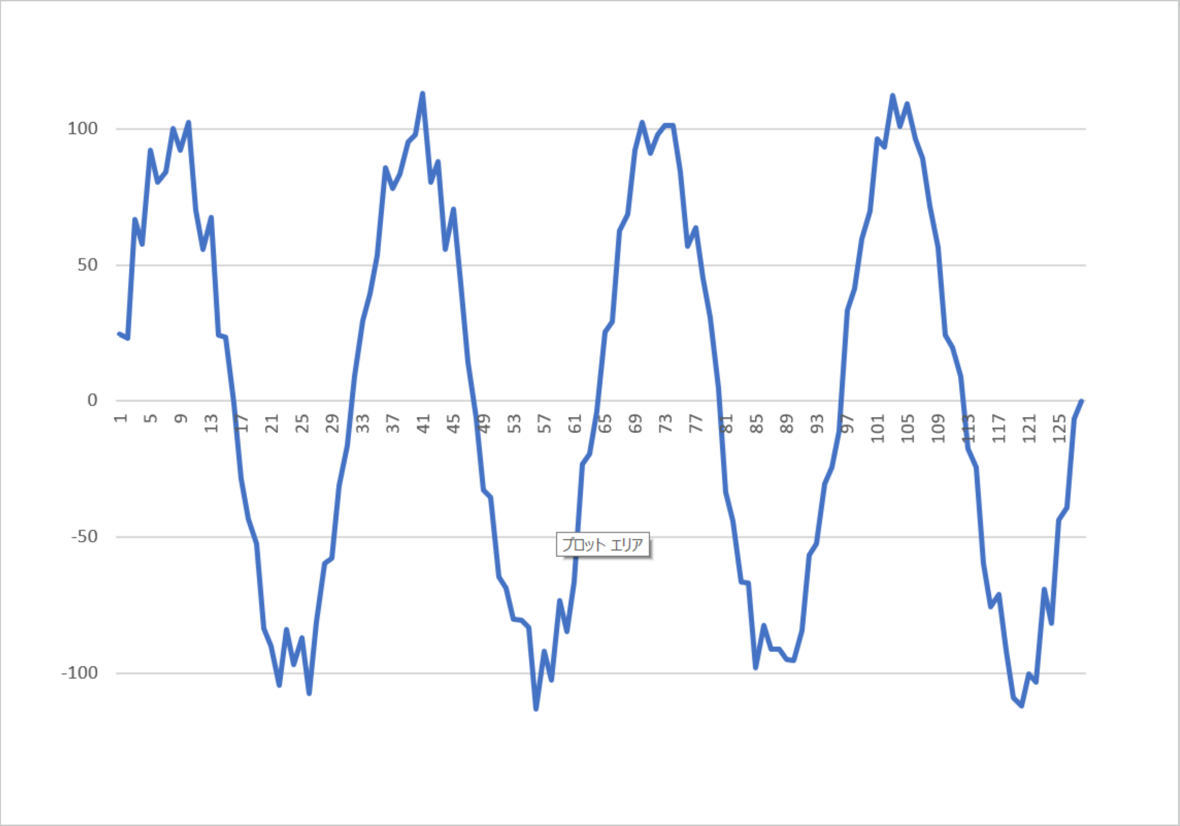
$$g(m,n) = \begin{cases} 1, & m-n=1\ \text{のとき} \\ 3, & m-n=0\ \text{のとき} \\ 5, & m-n=-1\ \text{のとき} \\ 0, & \text{上記以外} \end{cases}$$

$$y(m) = \{-,23,44,62,-\}$$

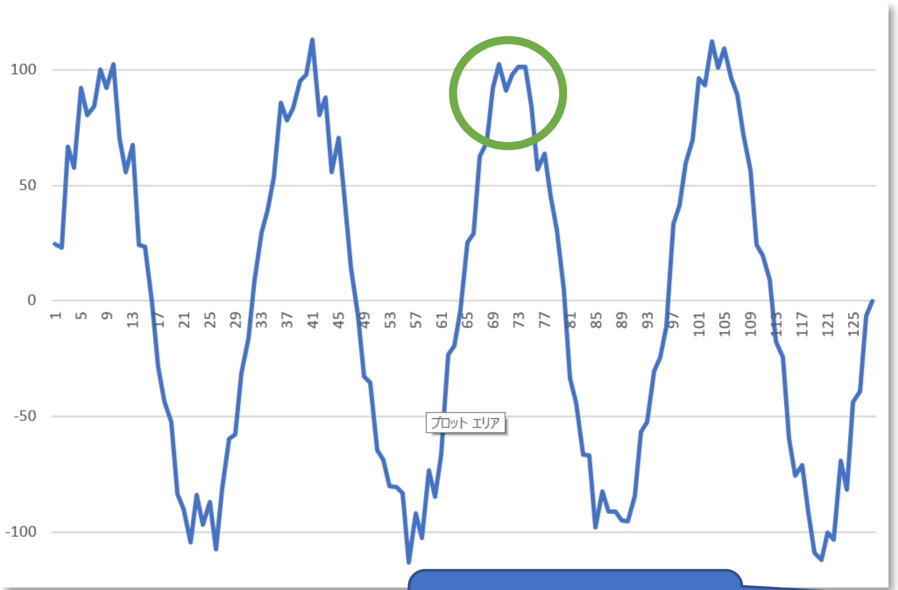
空間的の意

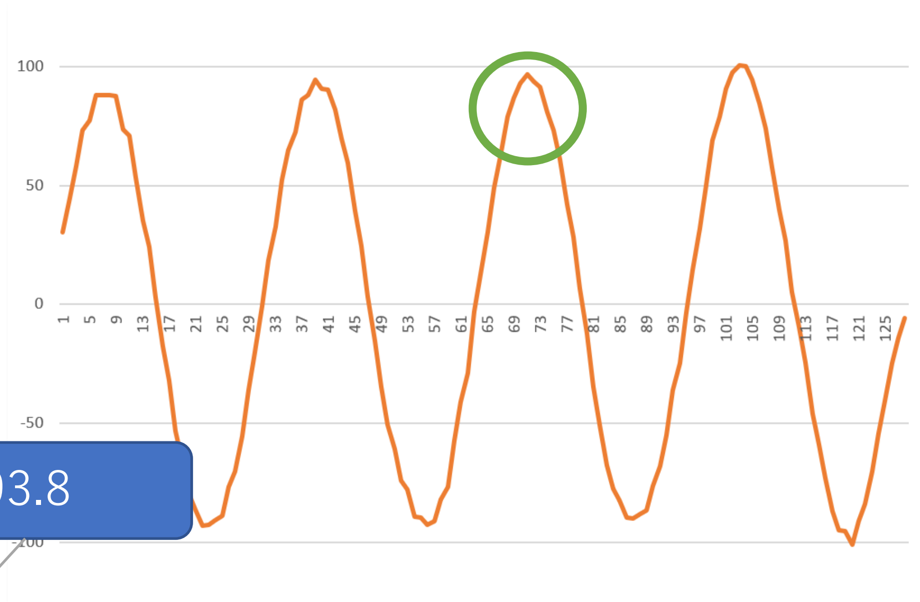
Spatialな場合

ノイズが目立たなくなり、形がはっきりする！



Spatialな場合（どんな手順で？）

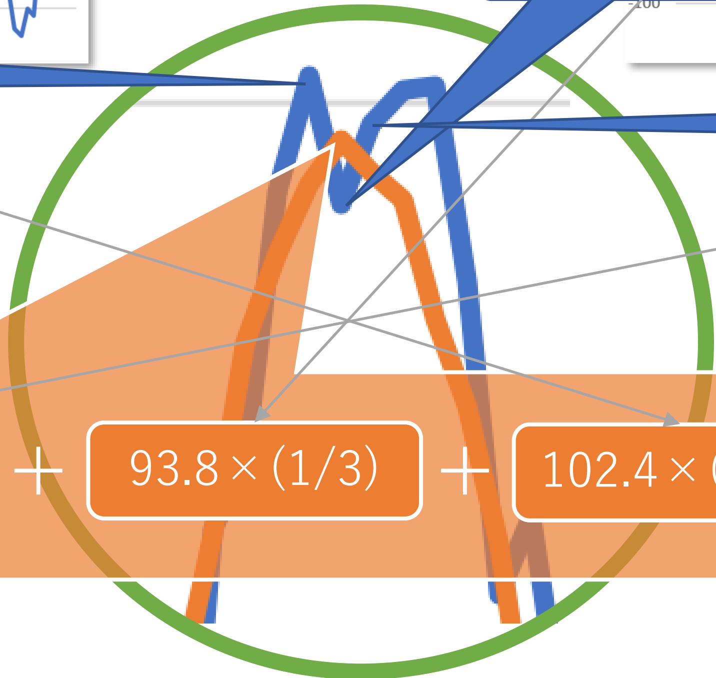




93.8

102.4

96.7



このような空間的な操作を
繰り返していく

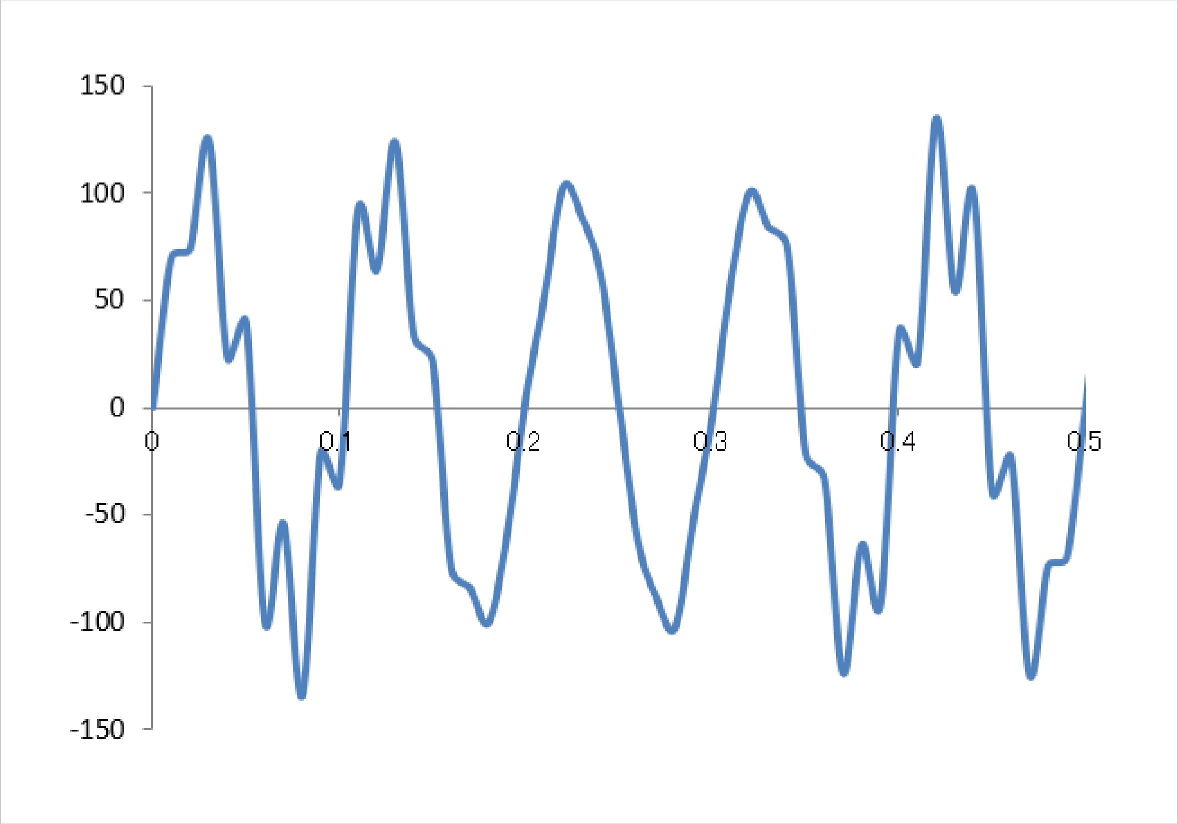
96.7×(1/3) ＋ 93.8×(1/3) ＋ 102.4×(1/3) ＝ 97.6

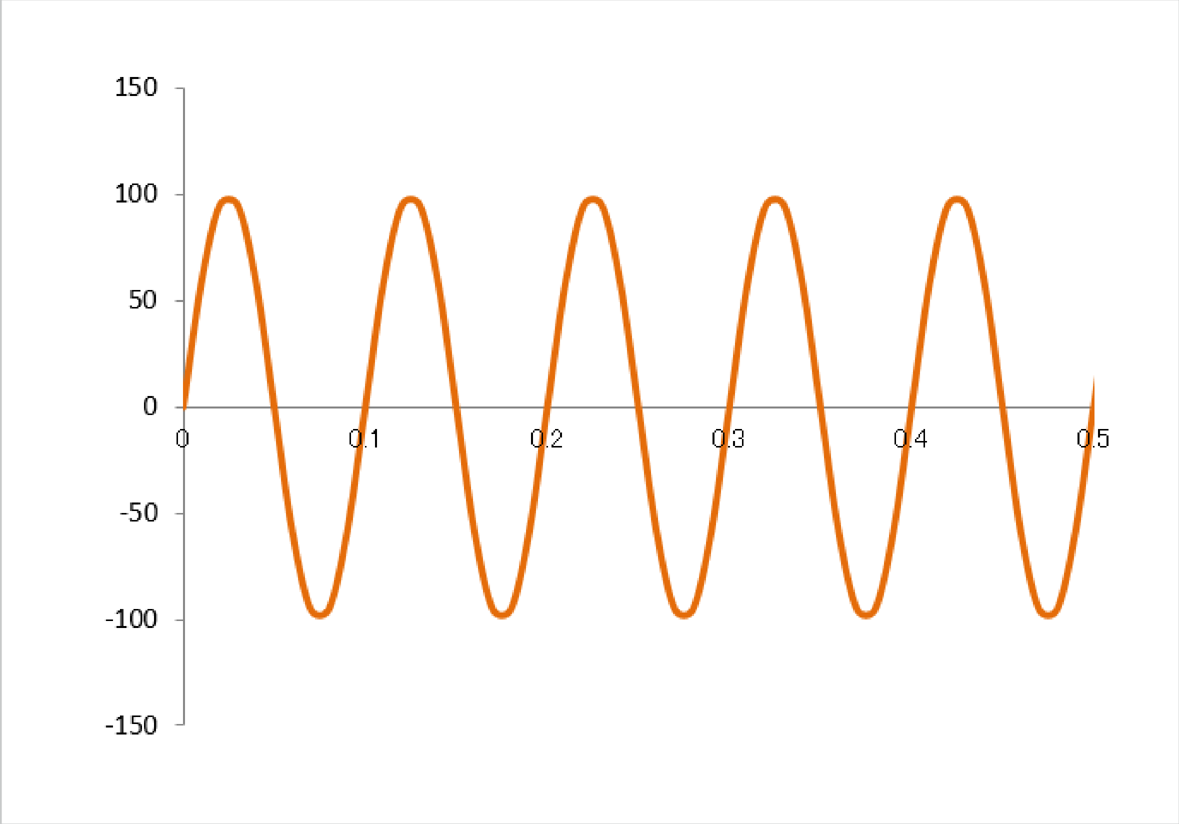
重みの係数は用途に
合わせて変更可能

スペクトル的の意

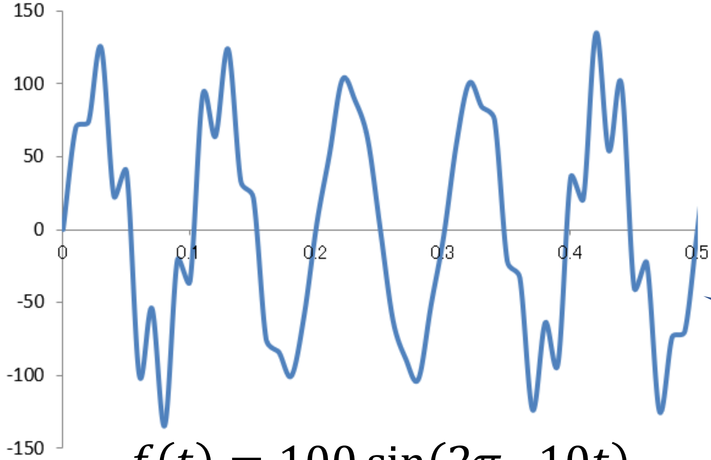
Spectralな場合

スペクトルに分解することで、特徴的な成分が明らかに！





Spectralな場合（どんな手順で？）



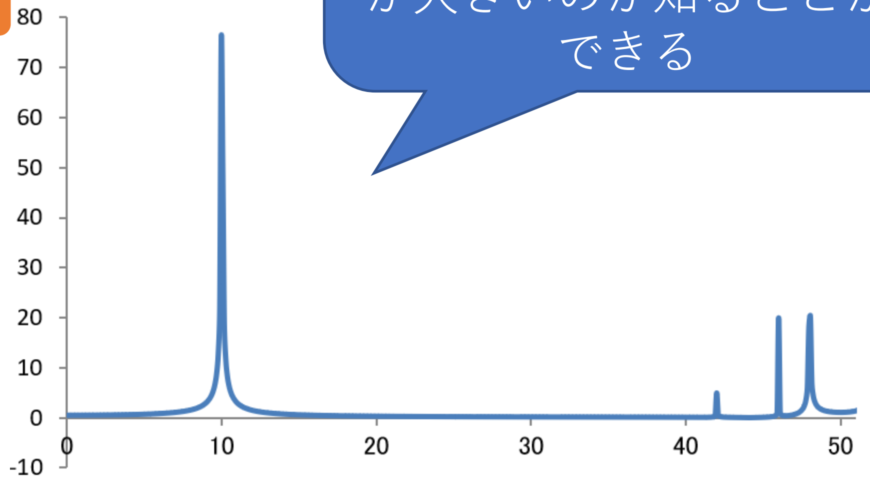
$$f(t) = 100\sin(2\pi \cdot 10t) \\ +5\sin(2\pi \cdot 42t) \\ +20\sin(2\pi \cdot 46t) \\ +30\sin(2\pi \cdot 48t)$$

複雑な関数も（様々な周波数の）
三角関数などの足し合わせとして
表現できる！

（離散）フーリエ変換

重み

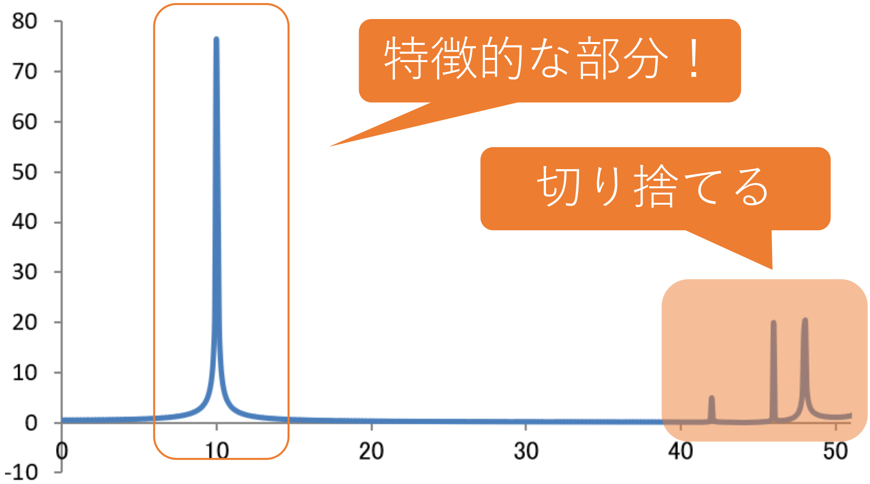
$$F(t) = \sum_{x=0}^{N-1} f(x) \boxed{\exp\left(-i\frac{2\pi t x}{N}\right)}$$



これが「スペクトル」！

どの周波数の成分の強度が大きいのか知ることができる

注目部分を取り出す

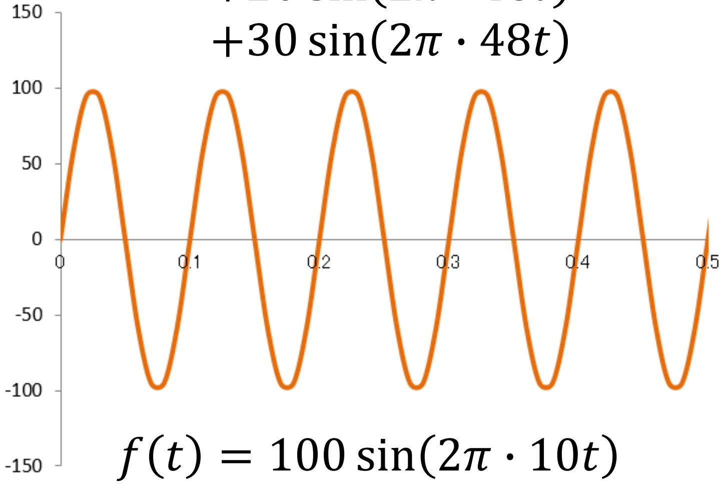


特徴的な部分！

切り捨てる

逆（離散）フーリエ変換

$$f(x) = \frac{1}{N}\sum_{t=0}^{N-1} F(t)\exp\left(i\frac{2\pi x t}{N}\right)$$



$$f(t) = 100\sin(2\pi \cdot 10t)$$

ここまでのまとめ

- 畳み込み→フィルターをかけること
- Spatialな場合→ノイズを目立たなくさせ、特徴をはっきり
- Spectralな場合→スペクトルに分解することで、特徴をはっきり

- これを2次元画像に用いる→CNN
- これを「グラフ」に用いる→GCN

- 大雑把に知りたいだけならば、ここまでOK！

Spatial GCN

この章の流れ

- 理解の前提となる事柄を解説する。

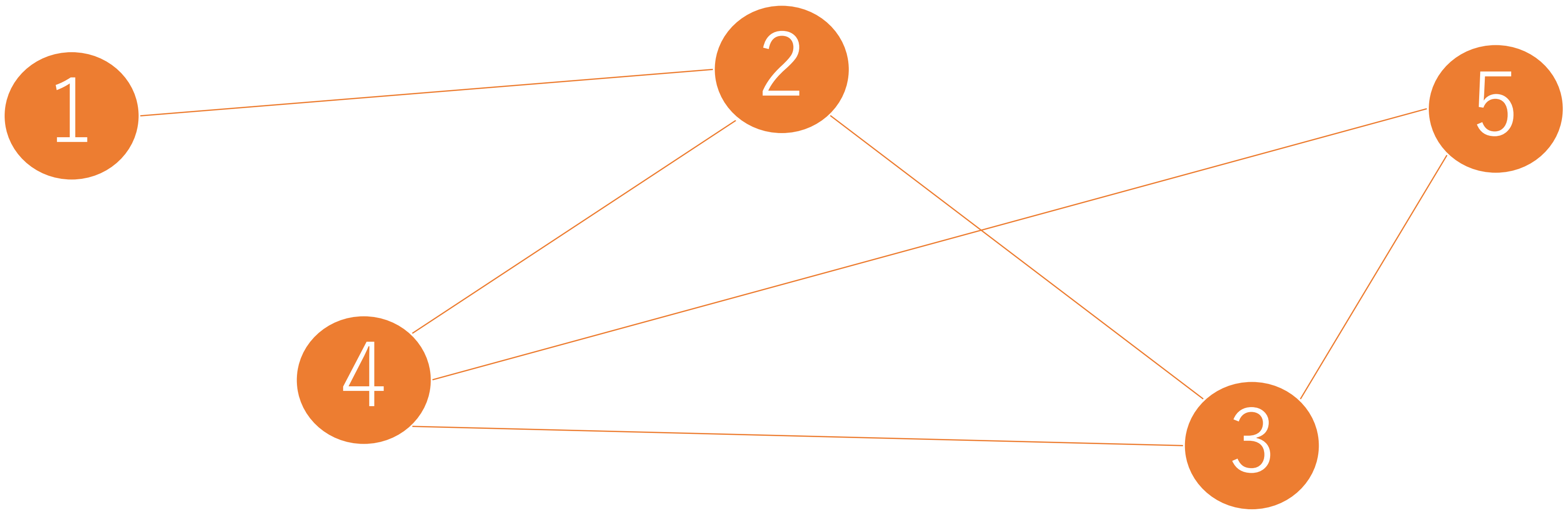
「グラフ」について簡単に説明

- J.Brunaらが2014年に発表した
”Spectral Networks and Deep Locally Connected Networks on Graphs“
の内容をもとにしてSpatial GCNを紹介する。

「グラフ畳み込み」を定式化した最初期の論文（無料で読める！）

- ざっくりと式の流れを追いかける。
- 計算コストについて簡単に考察する。

「グラフ」って何？



- 一言でいうと「ネットワーク」のこと
- ノード（頂点）とノードとがエッジ（辺）で結ばれたデータ構造
- 直線状の構造や格子状の構造に比べて構造が自由
- 様々に応用できる

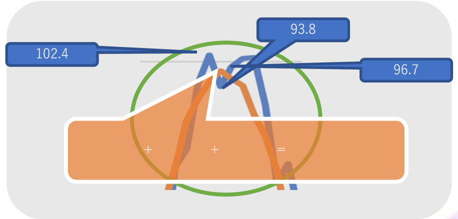
例）「人間関係」「文中の単語の共起関係」「分子構造」「交通状況」など

Spatial GCNの概要

クラスターに分割する方法は様々ある。今回はあまり議論せず「凝集型」と呼ばれる方法を用いているらしい

局所性（「近くにあるノード」の決め方）

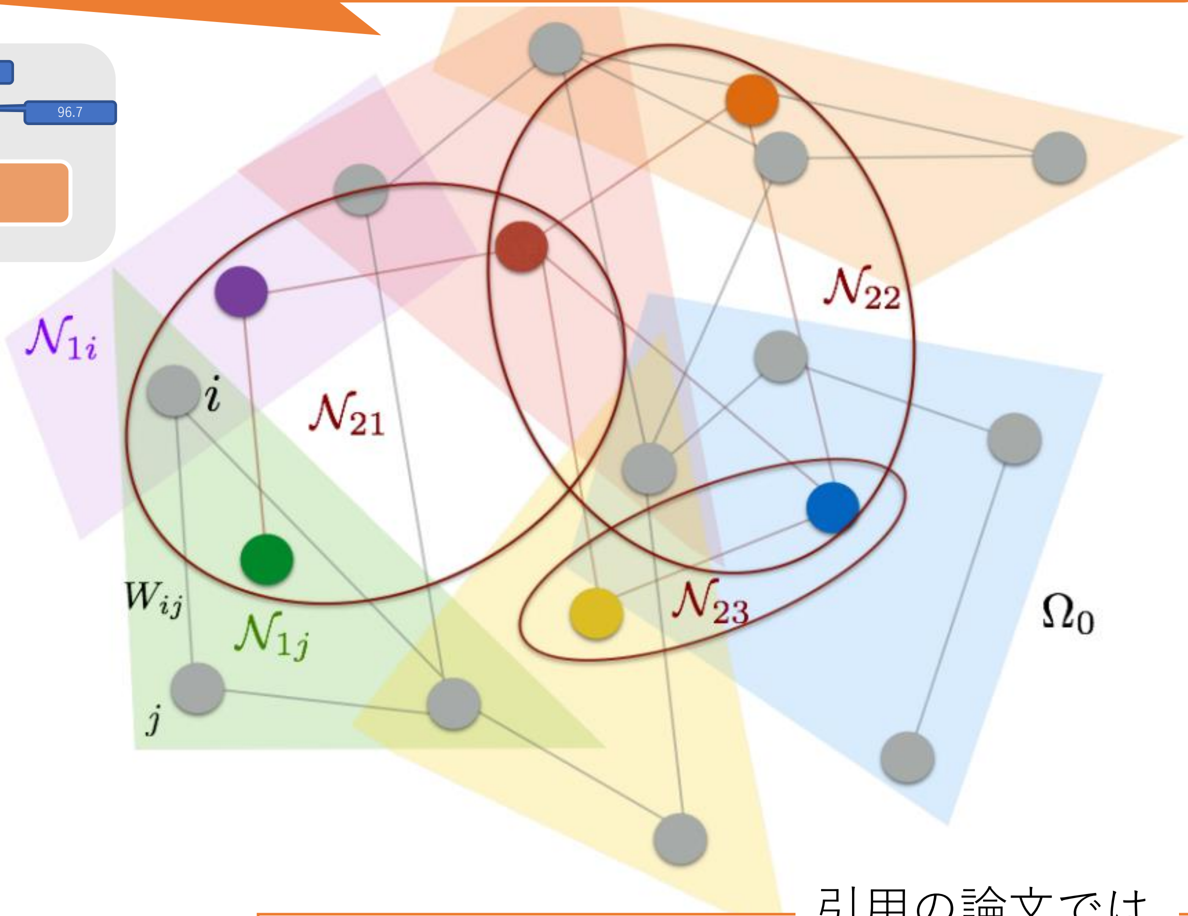
$$N_\delta(j) = \{i \in \Omega : W_{ij} > \delta\}$$



jの近くにあるノード（頂点）

それはiである（そしてiはノードの集合Ωに含まれている）

このiを決める基準は、「W_{ij}が閾値であるδよりも大きいとき」というものである。
（ちなみにW_{ij}はiとjを指定することで1つの値に決定する）



隣接関係（分割したクラスター内の要素同士の関係）

$$\mathcal{N}_k = \{\mathcal{N}_{k,i}\ ;\ i = 1 \ldots d_{k-1}\}$$

第k層における、各クラスター内の要素同士の関係は

kとiとを指定することで知ることができる

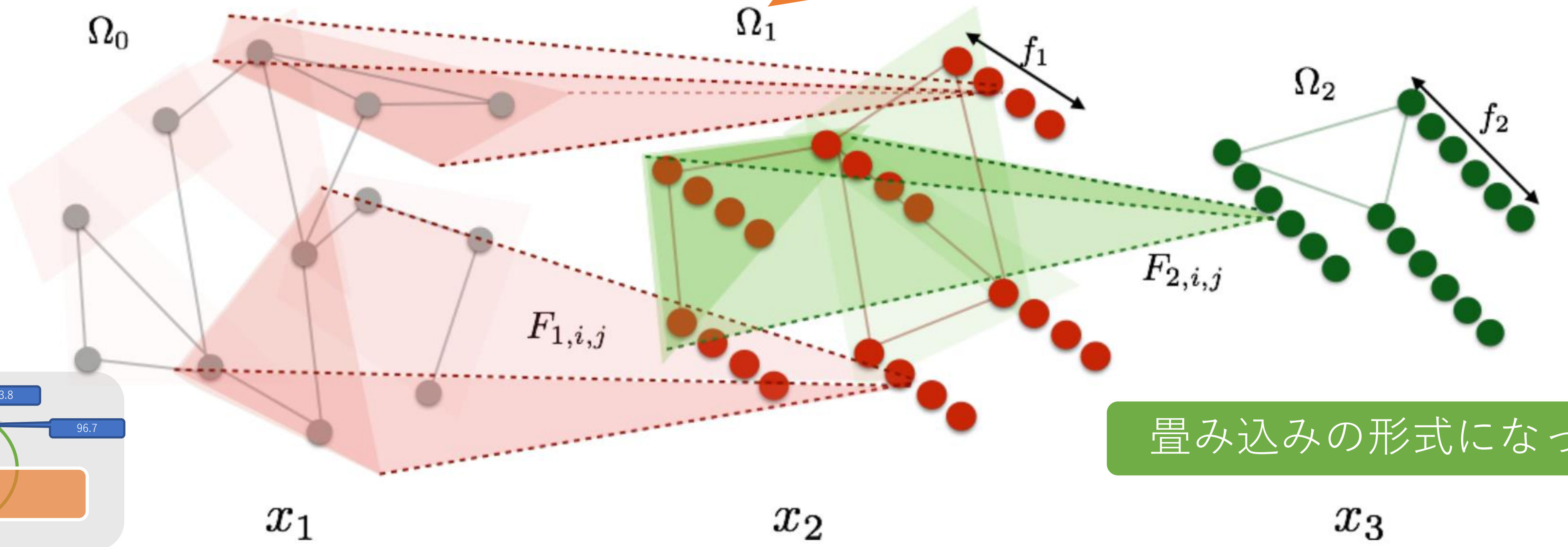
iは分割したクラスターの第i番目ということ

引用の論文では

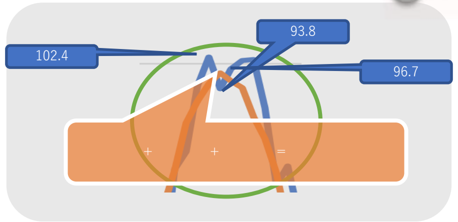
$$\begin{aligned}
W_0 &= W \\
A_k(i,j) &= \sum_{s\in\Omega_k(i)} \sum_{t\in\Omega_k(j)} W_{k-1}(s,t)\ ,\ (k \le K) \\
W_k &= \mathrm{rownormalize}(A_k)\ ,\ (k \le K) \\
\mathcal{N}_k &= \mathrm{supp}(W_k)\ .\ (k \le K)
\end{aligned}$$

Spatial GCNの概要

Ω_kは、Ω_{k-1}をd_k個のクラスターに分割したもの



畳み込みの形式になっている！



プーリングの機能

活性化関数

第k層目の入力

第k層目のフィルターの数

第k層目の出力
＝第$k+1$層目の入力

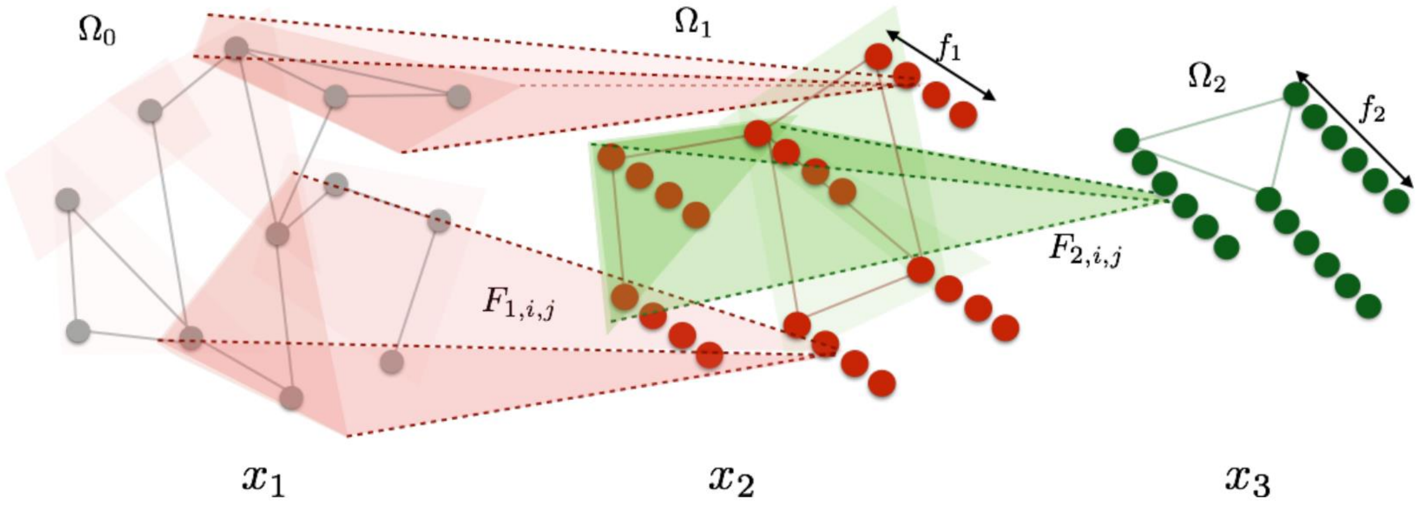
$$x_{k+1,j} = L_k h\left(\sum_{i=1}^{f_{k-1}} F_{k,i,j} x_{k,i}\right) \quad (j = 1 \ldots f_k)$$

フィルター（隣接関係$\mathcal{N}_k$のあるところだけ値を持つ「すかすか」の行列）

Spatial GCNの計算コスト

- 入力するべきパラメーターが多いほど計算コストは大きくなっていく傾向がある！

- そこでここではパラメーター数と計算コストを比例しているものとみなす



$$S_k \cdot |\Omega_k| \approx \alpha \cdot |\Omega_{k-1}|$$

$$x_{k+1,j} = L_k h\left(\sum_{i=1}^{f_{k-1}} F_{k,i,j} x_{k,i}\right) \quad (j = 1 \ldots f_k)$$

$$O(S_k \cdot |\Omega_k| \cdot f_k \cdot f_{k-1}) = O(n)$$

隣接ノード数の平均（$\mathcal{N}_k$から導く）

「nのオーダー」と読む（nに比例する程度の大きさという意味）

Spectral GCN

この章の流れ

- 理解の前提となる事柄を解説する。

「グラフ」のフーリエ変換について議論
（少々長い！それでもざっくりとした説明。
「スペクトル解析」や「グラフ理論」につ
いて学習すると理解が深まる！）

- J.Brunaらが2014年に発表した
  ”Spectral Networks and Deep Locally Connected Networks on Graphs“
  の内容にも触れつつ
  Ziwei Zhangらが2020年に発表した
  “Deep Learning on Graphs: A Survey”
  の内容をもとにSpectral GCNを紹介する。

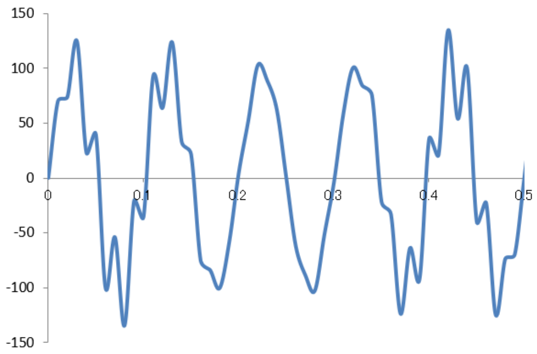
様々な手法を比較したサーベイ論文
（これも無料で読める！）

- ざっくりと式の流れを追いかける。

- 計算コストについて簡単に考察する。

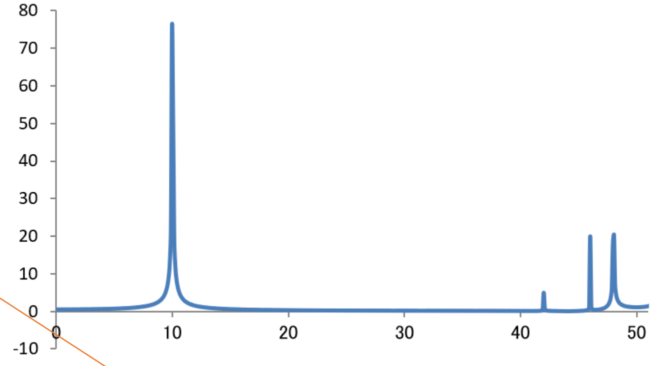
「グラフ」をフーリエ変換するには？

1次元の場合を思い出す…

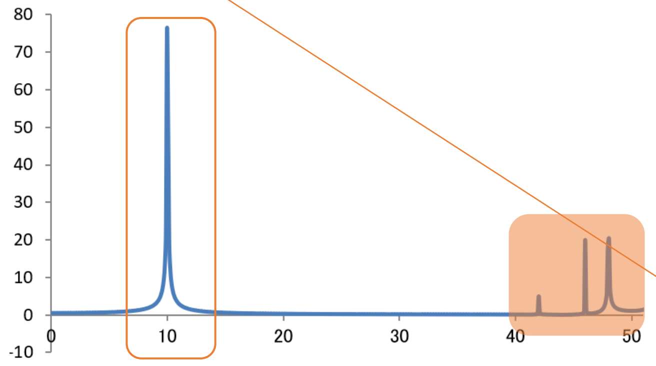


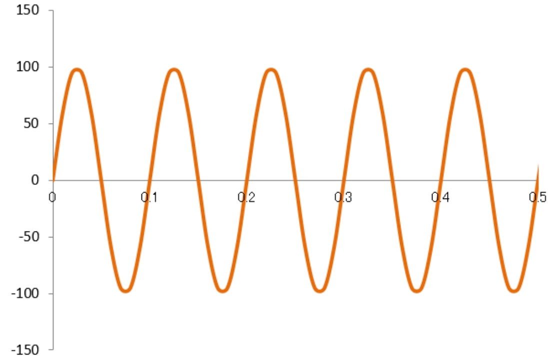
（離散）フーリエ変換

$$F(t) = \sum_{x=0}^{N-1} f(x) \boxed{\exp\left(-i\frac{2\pi t x}{N}\right)}$$



注目部分を取り出す





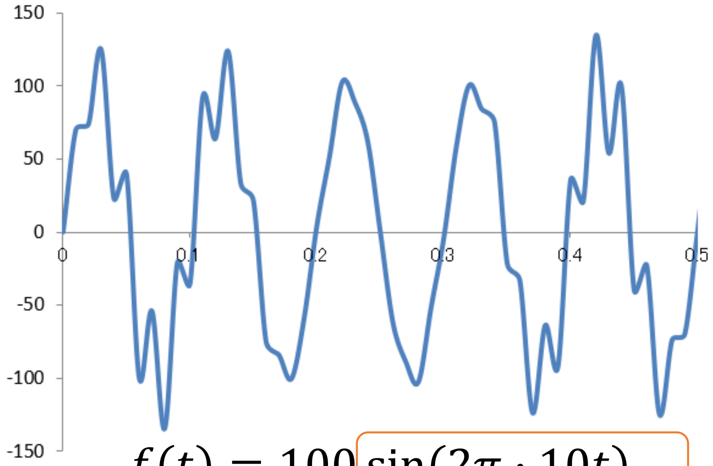
逆（離散）フーリエ変換

$$f(x) = \frac{1}{N} \sum_{t=0}^{N-1} F(t) \boxed{\exp\left(i\frac{2\pi x t}{N}\right)}$$

重み

- 1次元の場合の「重み」関数を、「グラフ」の場合に拡張するには？

「グラフ」をフーリエ変換するには？

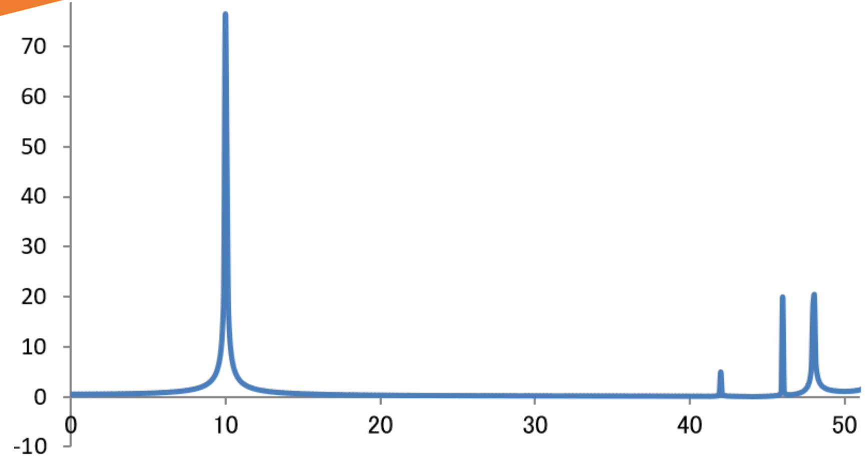


$$f(t) = 100\,\boxed{\sin(2\pi \cdot 10t)}$$
$$+5\sin(2\pi \cdot 42t)$$
$$+20\sin(2\pi \cdot 46t)$$
$$+30\sin(2\pi \cdot 48t)$$

（離散）フーリエ変換

$$F(t) = \sum_{x=0}^{N-1} f(x)\,\boxed{\exp\left(-i\frac{2\pi t x}{N}\right)}$$

$$e^{i\theta} = \cos\theta + \sin i\theta$$



$$\sin\theta = \frac{e^{i\theta} - e^{-i\theta}}{2i}$$

$$\frac{d^2}{dt^2}\sin\omega t = -\omega^2 \sin\omega t$$
$$\frac{d^2}{dt^2}e^{i\omega t} = -\omega^2 e^{i\omega t}$$

行列の固有値・固有ベクトル
の関係に似ている！

- 三角関数と指数関数は親戚
- もとの関数は「重み」関数の足し合わせでできている
- 「重み」関数の特徴　2階微分すると元にもどること

「グラフ」をフーリエ変換するには？

2階微分の性質とは？

$$\frac{d}{dx}f = \lim_{\Delta x \to 0} \frac{f(x+\Delta x) - f(x)}{\Delta x}$$

$$\frac{d^2}{dx^2}f = \lim_{\Delta x \to 0} \frac{\frac{f(x+\Delta x+\Delta x) - f(x+\Delta x)}{\Delta x} - \frac{f(x+\Delta x) - f(x)}{\Delta x}}{\Delta x}$$

$$= \lim_{\Delta x \to 0} \frac{f(x+\Delta x+\Delta x) - f(x+\Delta x)}{(\Delta x)^2} - \frac{f(x+\Delta x) - f(x)}{(\Delta x)^2}$$

$$= \lim_{\Delta x \to 0} \frac{\boxed{f(x+2\Delta x)} - \boxed{2f(x+\Delta x)} + \boxed{f(x)}}{(\Delta x)^2}$$

自分自身と

その両隣との差

$$\frac{d^2}{dt^2}\sin\omega t = -\omega^2 \sin\omega t$$

$$\frac{d^2}{dt^2}e^{i\omega t} = -\omega^2 e^{i\omega t}$$

行列の固有値・固有ベクトルの関係に似ている！

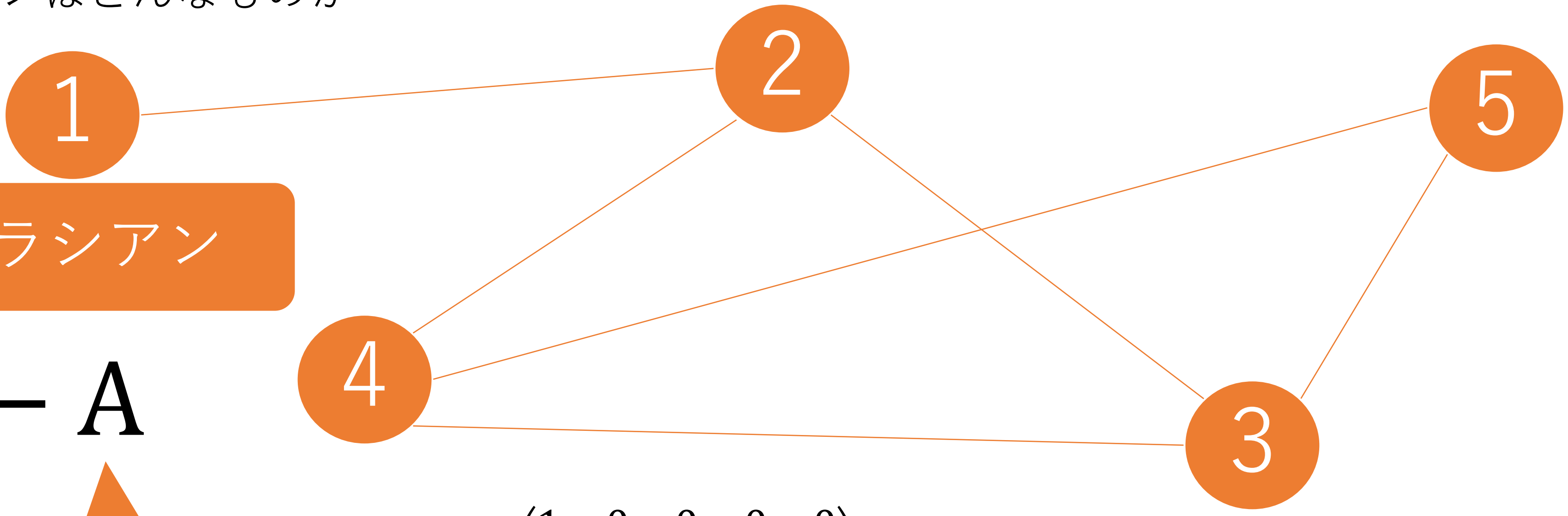
$$= \lim_{\Delta x \to 0} \frac{f(x+\Delta x) - 2f(x) + f(x-\Delta x)}{(\Delta x)^2}$$

と表現しても同様のこと

- 2階微分の性質
- 「グラフ」における2階微分とは？→自分自身とその周辺との差！ →グラフラプラシアン
- グラフラプラシアンの固有ベクトルを「重み」関数にすればよい！

「グラフ」をフーリエ変換するには？

グラフラプラシアンはどんなものか



グラフラプラシアン

$$L = D - A$$

次数行列「ノードから何本エッジが伸びているか」の情報

隣接行列「どのノードにつながっているか」の情報

$$D = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 2 \end{pmatrix}$$

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{pmatrix}$$

$$L = \begin{pmatrix} 1 & -1 & 0 & 0 & 0 \\ -1 & 3 & -1 & -1 & 0 \\ 0 & -1 & 3 & -1 & -1 \\ 0 & -1 & -1 & 3 & -1 \\ 0 & 0 & -1 & -1 & 2 \end{pmatrix}$$

「グラフ」をフーリエ変換するには？

1次元の場合を思い出す…

$$\frac{d^2}{dt^2}\sin\omega t = -\omega^2 \sin\omega t$$

$$\frac{d^2}{dt^2}e^{i\omega t} = -\omega^2 e^{i\omega t}$$

行列の固有値・固有ベクトル
の関係に似ている！

$$\mathrm{L}\vec{q} = \lambda\vec{q}$$

$$\mathrm{LQ} = \mathrm{Q}\Lambda$$

$$\mathrm{Q} = \begin{pmatrix} \vec{q_1} & \vec{q_2} & \cdots \end{pmatrix}$$

$$\vec{F} = \mathcal{F}_G\left(\vec{f}\right) = \mathrm{Q}^{\top}\vec{f}$$

$$\vec{f} = \mathcal{F}_G^{-1}\left(\vec{F}\right) = \mathrm{Q}\vec{F}$$

$$F(t) = \sum_{x=0}^{N-1} f(x) \boxed{\exp\left(-i\frac{2\pi t x}{N}\right)}$$

$$\vec{F} = \mathcal{F}_G\left(\vec{f}\right) = \sum_n \vec{f}\cdot\boxed{\overrightarrow{q_n}}$$

$$u_1 * u_2 = \sum_n u_1(n)\boxed{u_2(m,n)}$$

$$\mathcal{F}(u_1 * \boxed{u_2}) = \mathcal{F}(u_1)\boxed{\mathcal{F}(u_2)}$$

$$\mathcal{F}_G(u_1 \ *_G \boxed{u_2}) = \mathcal{F}_G(u_1) \odot \boxed{\mathcal{F}_G(u_2)}$$

$$u_1 * \boxed{u_2} = \mathcal{F}^{-1}\left(\mathcal{F}(u_1)\boxed{\mathcal{F}(u_2)}\right)$$

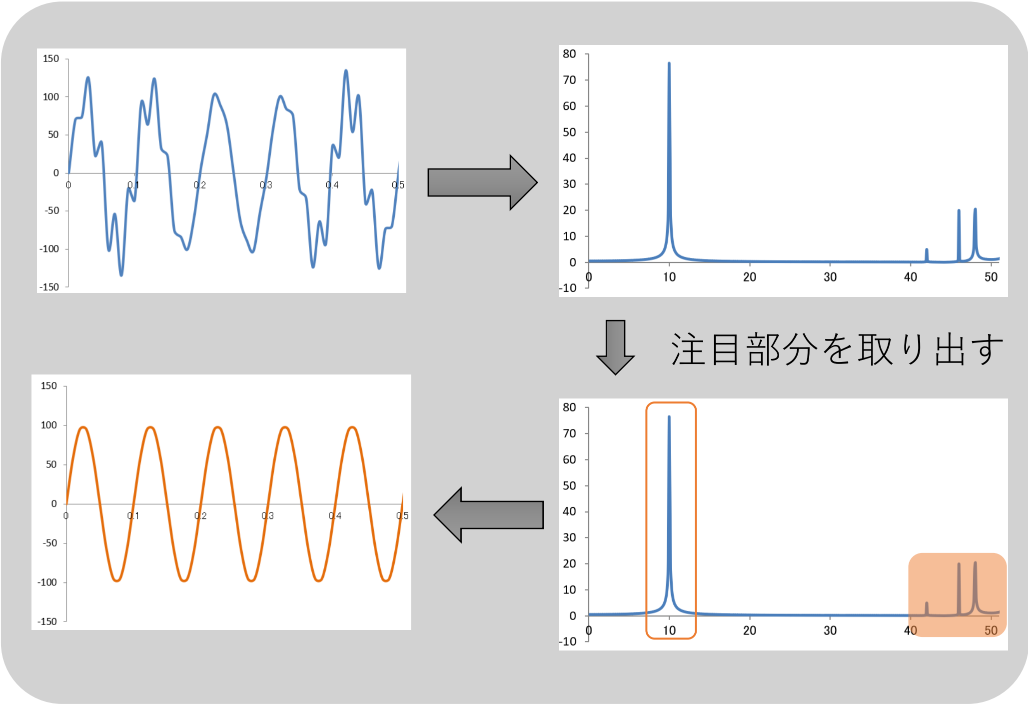
$$u_1 \ *_G \boxed{u_2} = \mathcal{F}_G^{-1}\left(\mathcal{F}_G(u_1) \odot \boxed{\mathcal{F}_G(u_2)}\right)$$

$$u_1 \ *_G \boxed{u_2} = \mathrm{Q}\left((\mathrm{Q}^{\top}u_1) \odot \boxed{(\mathrm{Q}^{\top}u_2)}\right)$$

Spectral GCNの概要

$$\mathbf{u}_1 *_G \mathbf{u}_2 = \boxed{\mathbf{Q}}\left(\boxed{\left(\mathbf{Q}^T\mathbf{u}_1\right)} \odot \boxed{\left(\mathbf{Q}^T\mathbf{u}_2\right)}\right)$$

$$\mathbf{u}' = \boxed{\mathbf{Q}}\boxed{\mathbf{\Theta}}\boxed{\mathbf{Q}^T\mathbf{u}}$$

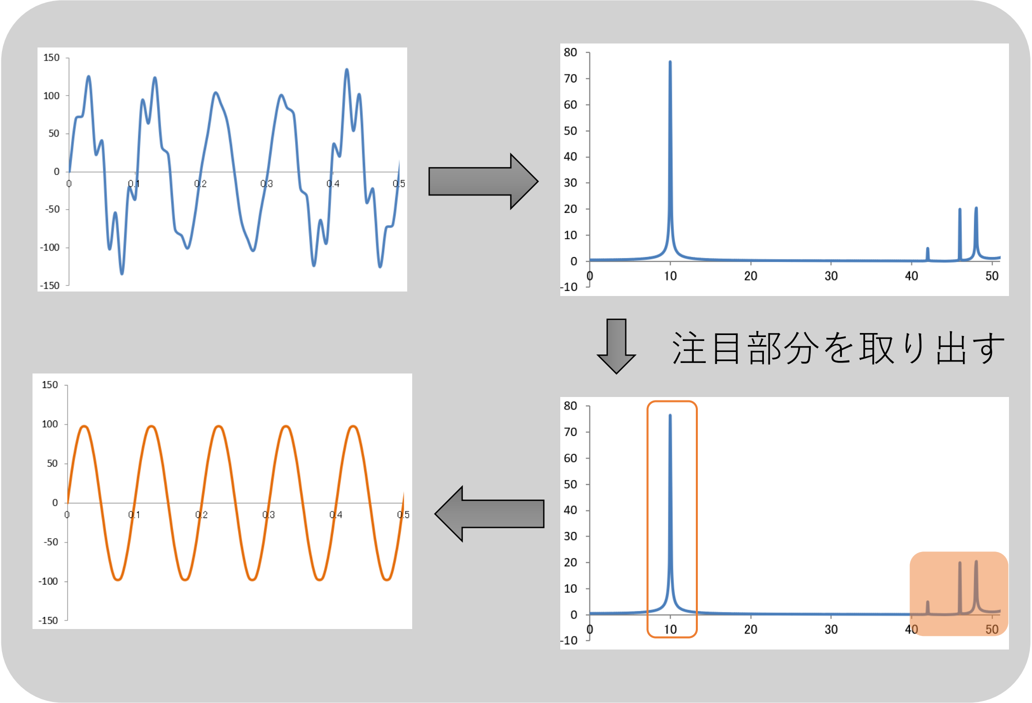


$$\mathbf{u}_j^{l+1} = \rho\left(\sum_{i=1}^{f_l} \boxed{\mathbf{Q}\mathbf{\Theta}_{i,j}^{l}\mathbf{Q}^T\mathbf{u}_i^{l}}\right) \quad j = 1, \ldots, f_{l+1}$$

活性化関数

Spectral GCNの計算コスト

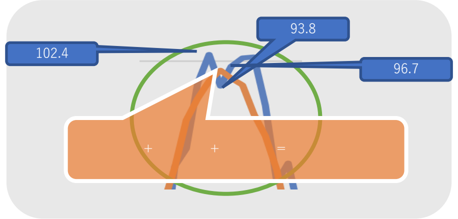
$$L\vec{q} = \lambda\vec{q}$$
$$LQ = Q\Lambda$$
$$L = Q\Lambda Q^{\mathsf{T}}$$



- 固有値分解をしなければならないので計算コストは高い
- 少なくとも$O(n^2)$　場合によっては$O(n^3)$

全体のまとめ

- 畳み込み→フィルターをかけること
- これを「グラフ」に用いる→GCN
- Spatial GCN
  「グラフ」を空間的に考えて畳み込む
- 計算コストは$O(n)$ほど



- Spectral GCN
  グラフをスペクトルに分解して畳み込む
- 計算コストは$O(n^2)$以上