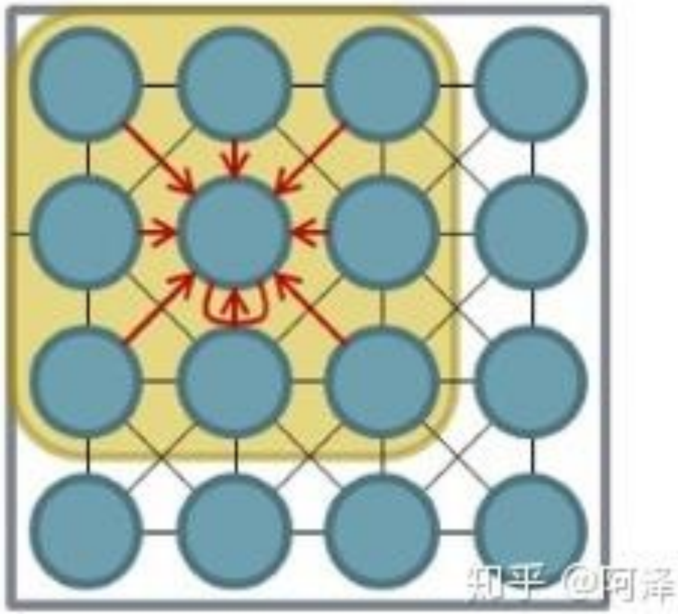
学术分享

GCN入门

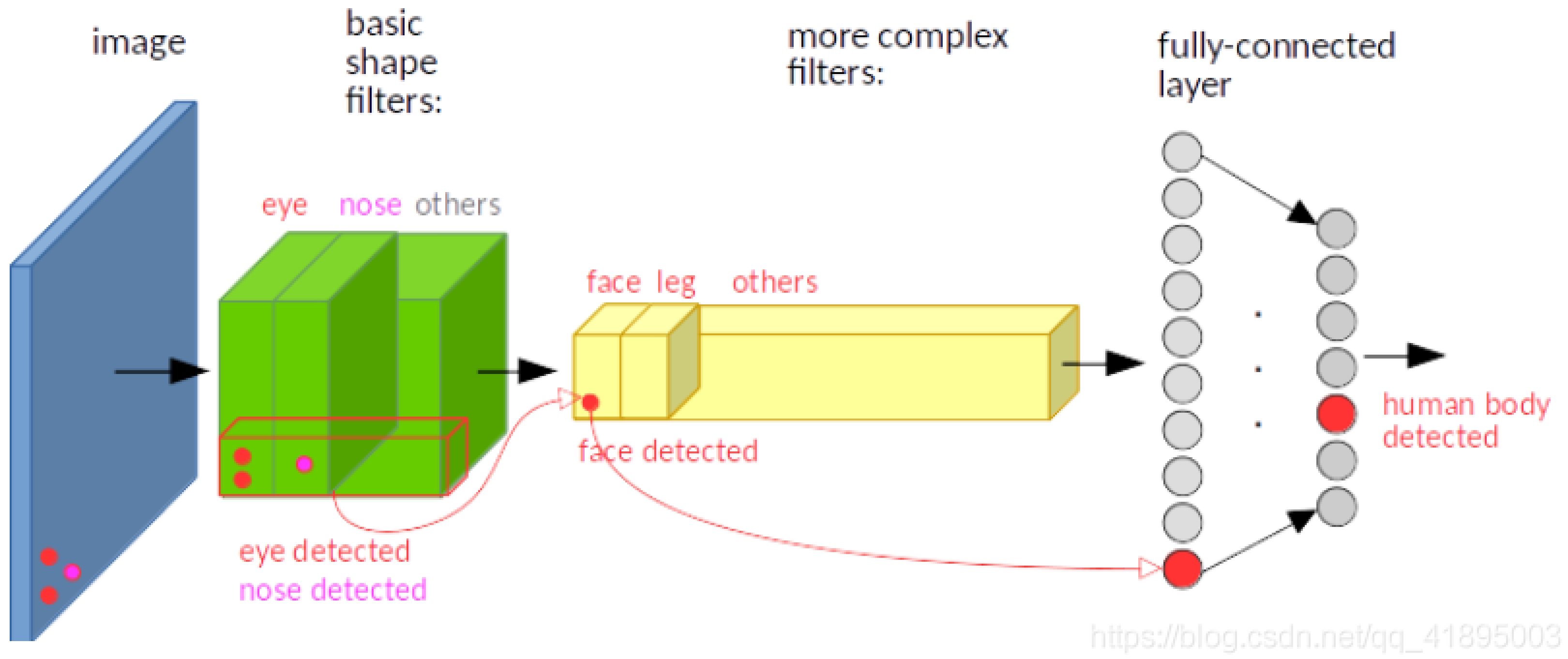
北京理工大学宇航学院

李嘉鑫

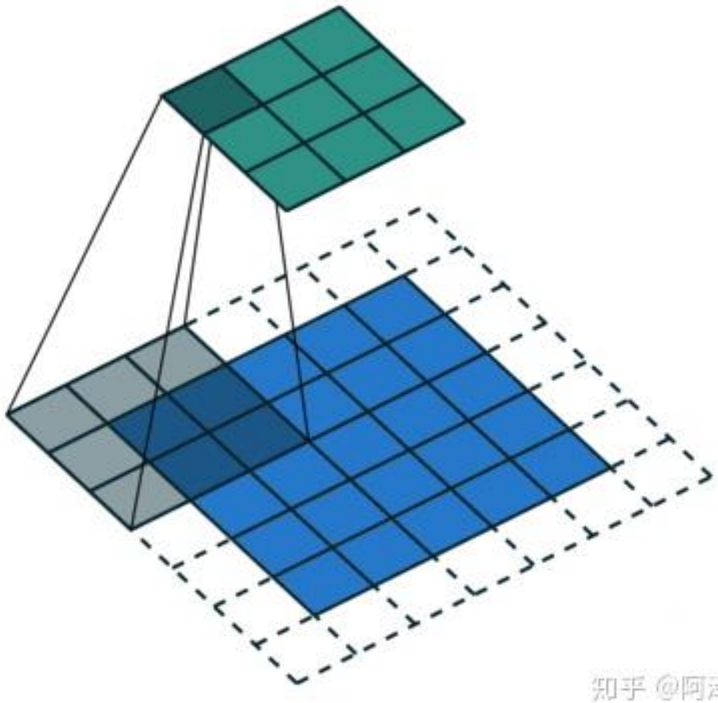
◆Convolutional Neural Network

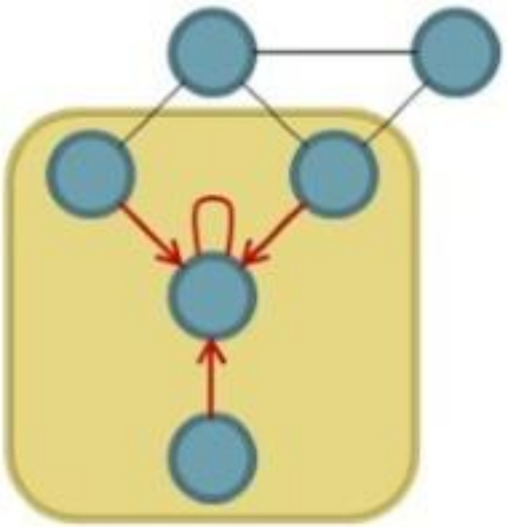


◆Local Translational Invariance

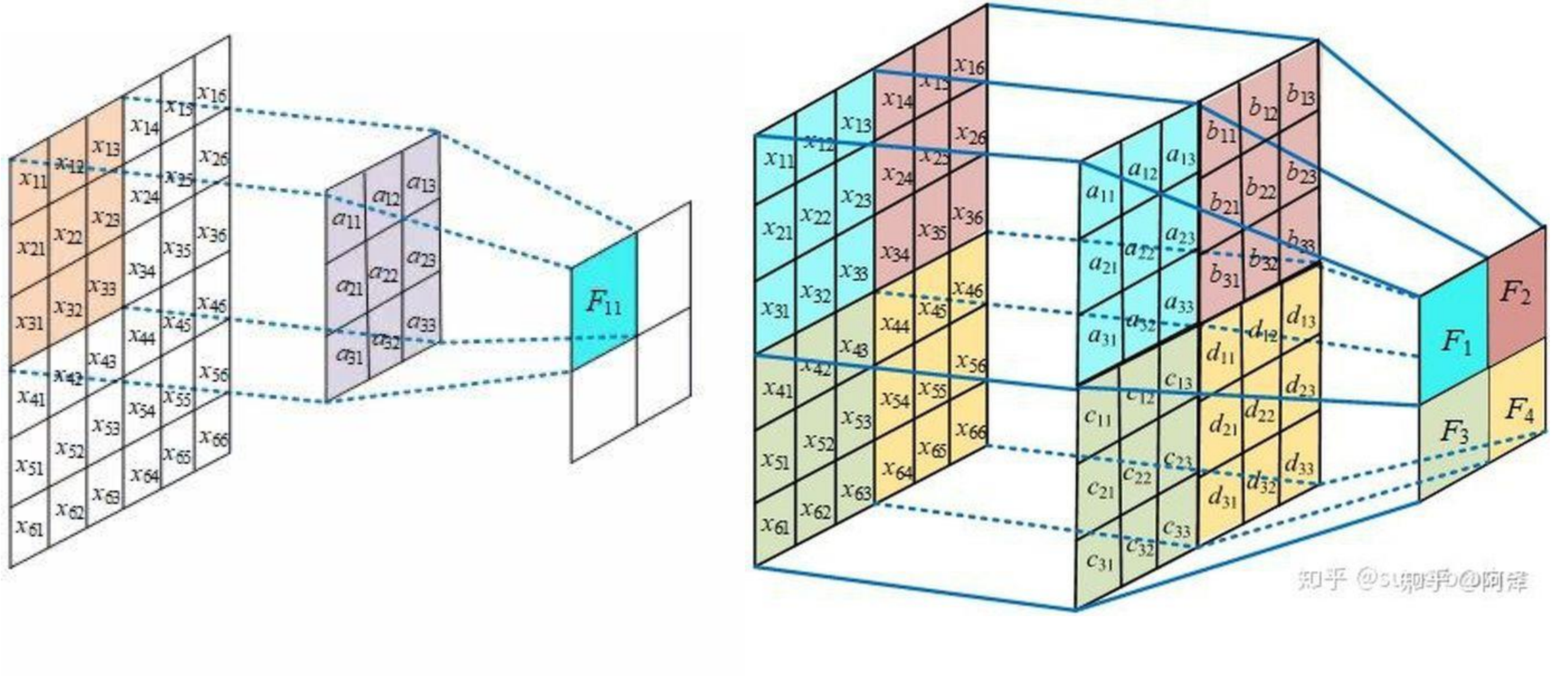


◆Local Translational Invariance

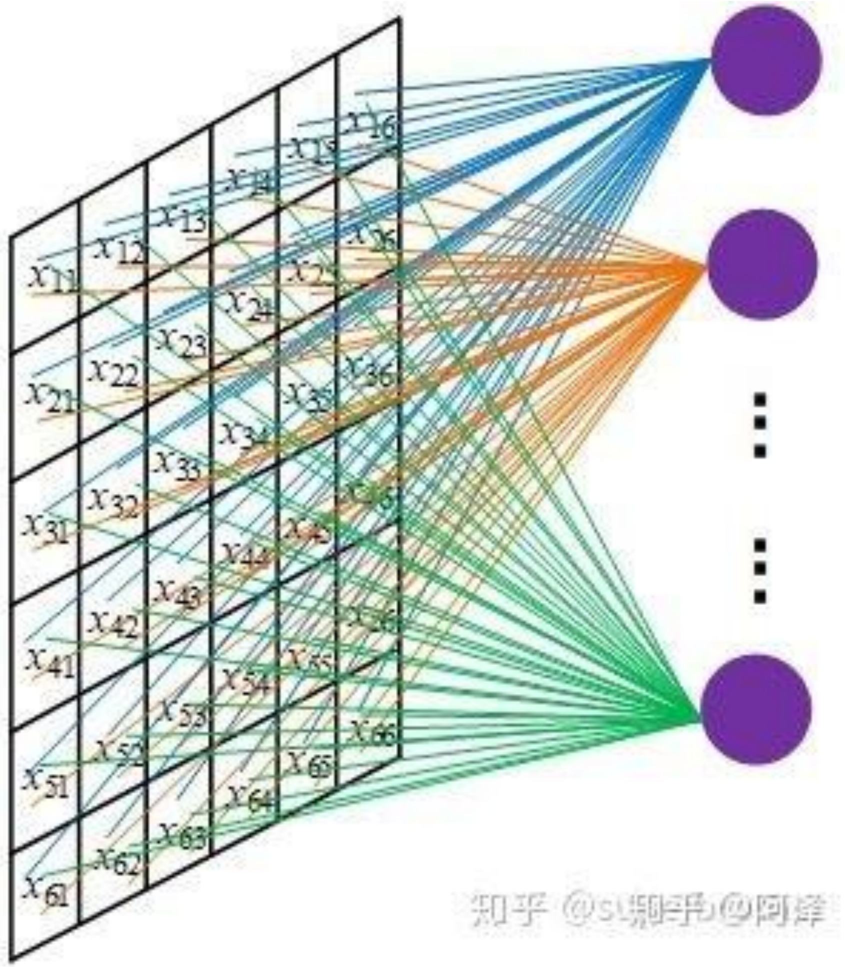




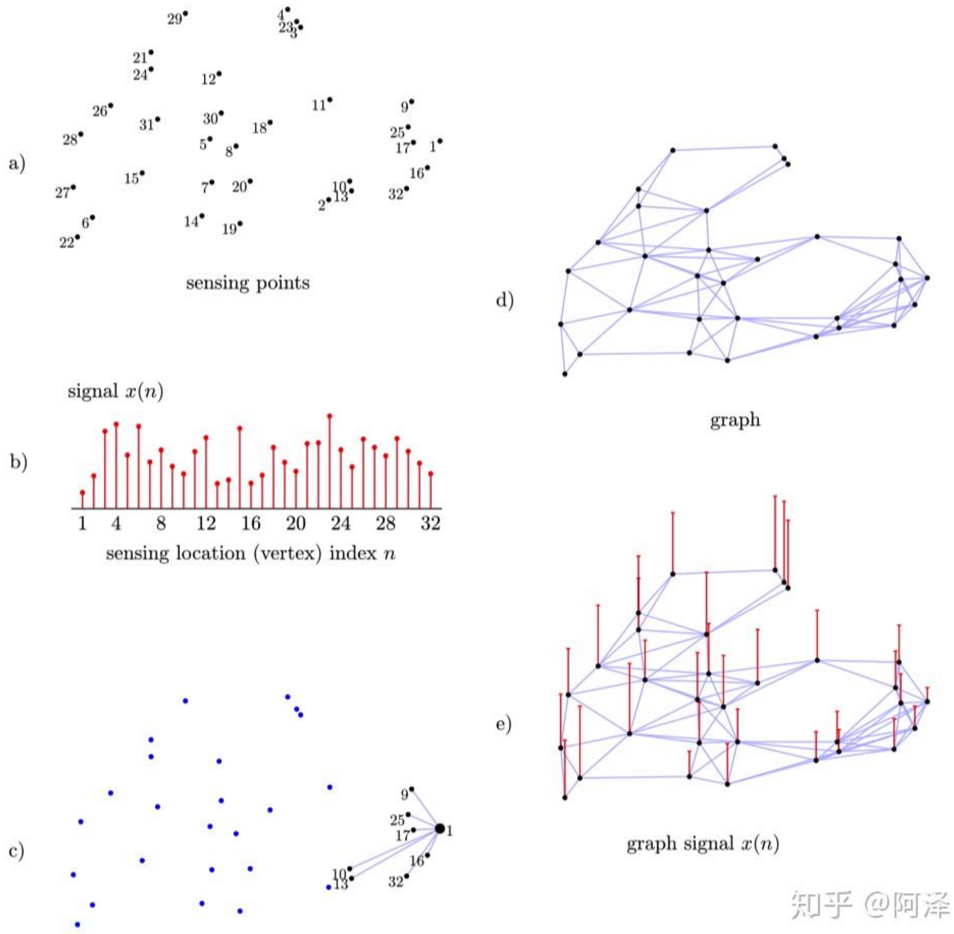
◆Convolution Kernels



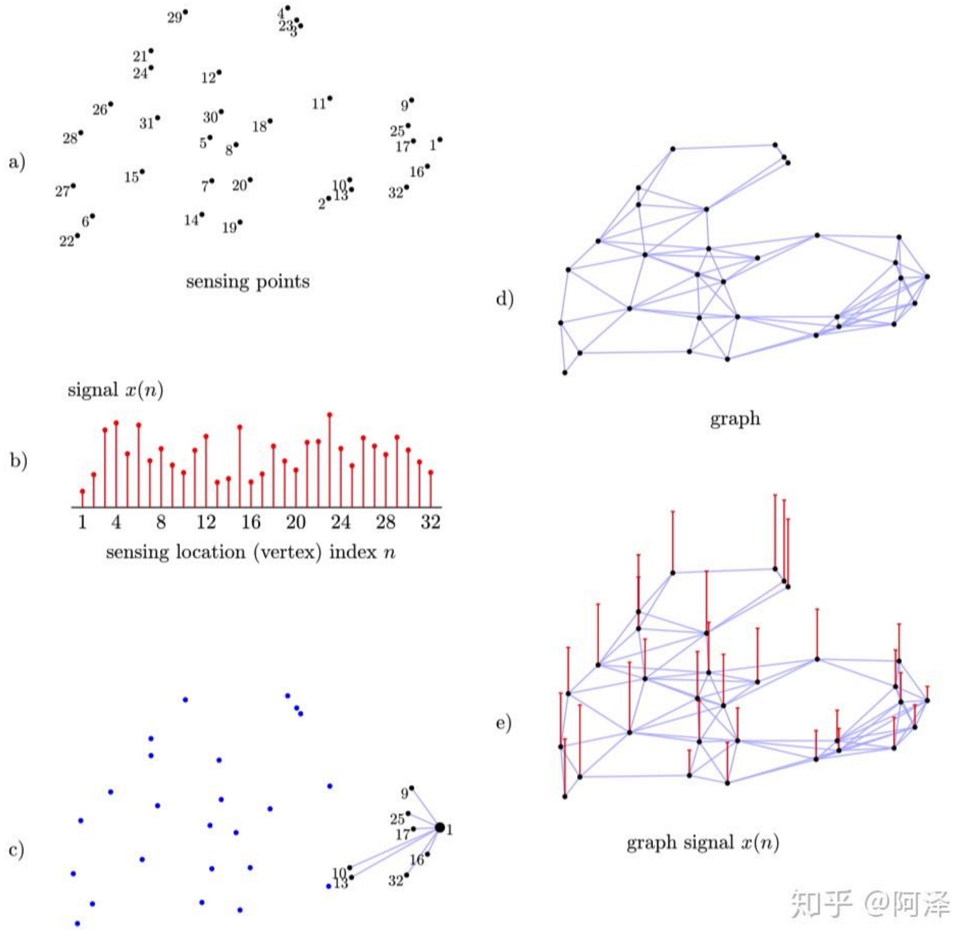
◆Convolution Kernels



◆图信号处理(Graph Signal Processing)



◆图信号处理(Graph Signal Processing)



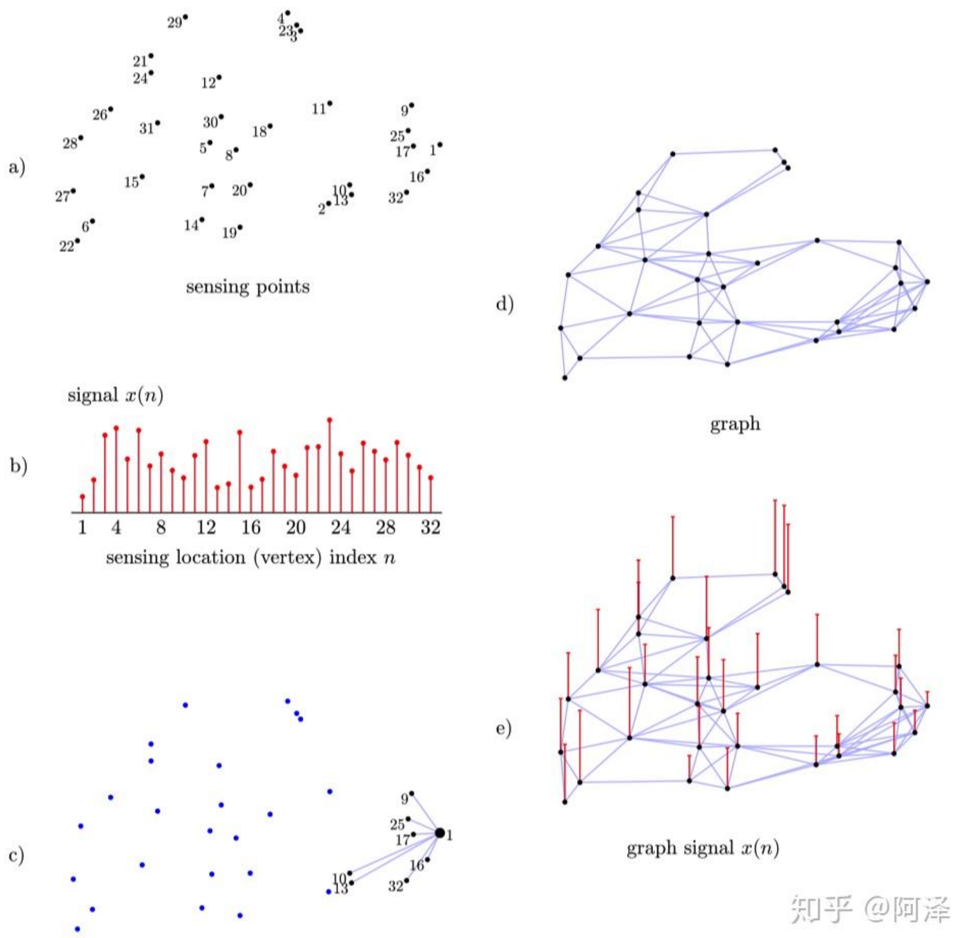
$$y(n) = x(n) + \sum_{m \in N(n)} x(m)$$

$$\mathbf{y} = \mathbf{x} + \mathbf{A}\mathbf{x}$$

$$\mathbf{y} = \mathbf{x} + \mathbf{W}\mathbf{x}$$

$$\mathbf{y} = \frac{1}{2}(\mathbf{x} + \mathbf{D}^{-1}\mathbf{W}\mathbf{x})$$

◆图信号处理(Graph Signal Processing)

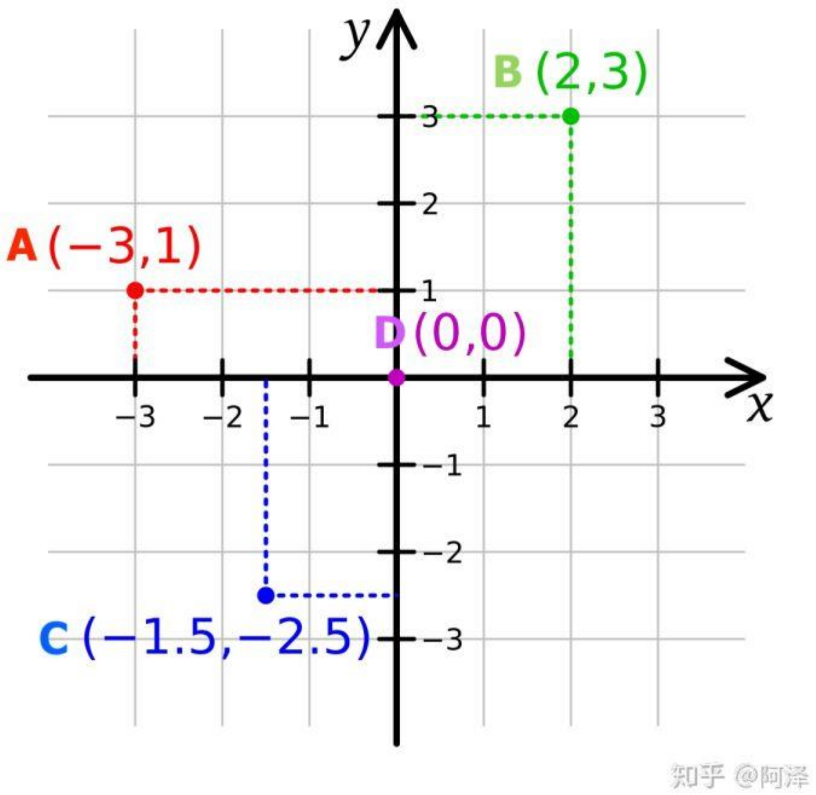


1.测量点构成节点（图 a），节点间的连通性和相关性构成边;
2.节点和边构成图（图 b），该图是信号域，表示测量信号的点以及它们之间的关系，并使用该图进行分析和处理;
3.测量温度是图的信号（图 e），这里的信号由真实温度和测量噪声所组成;
4.考虑测量位置，我们提出了局部平均和加权平均，这是最简单的图信号处理方式（Linear fist-order）。

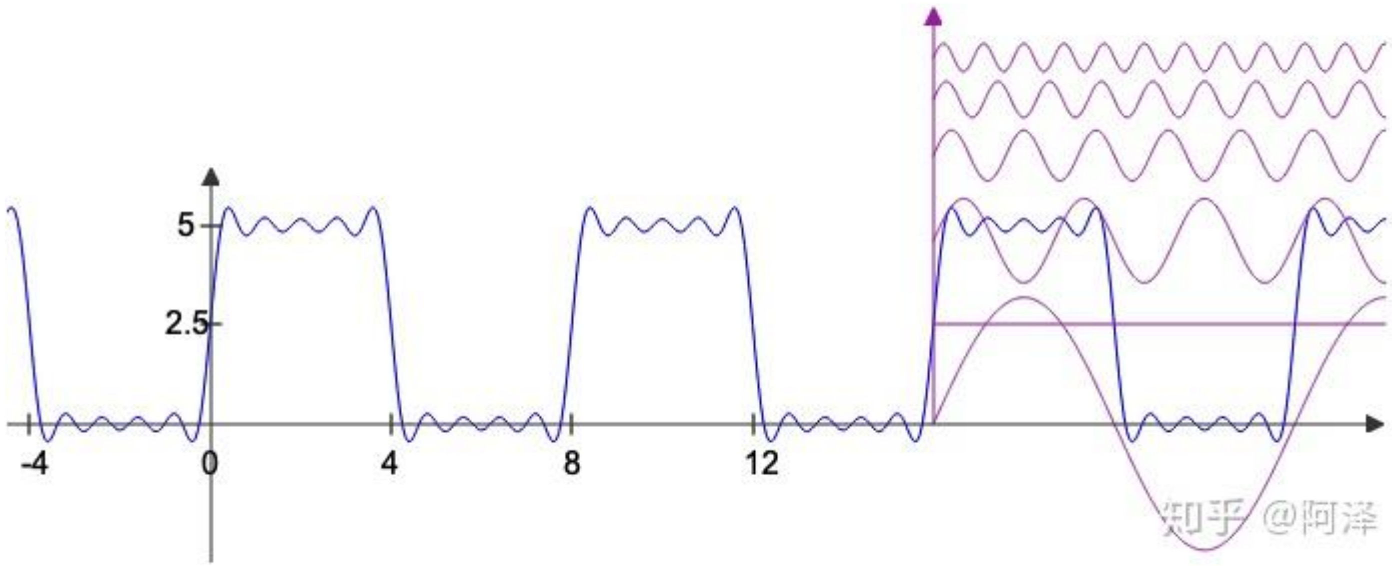
◆傅里叶变换（Fourier Transformer）

$$\hat{f}\left(\xi\right)=\int_{-\infty}^{\infty}f(x)e^{-2i\pi x\xi}\,dx$$

◆傅里叶变换（Fourier Transformer）

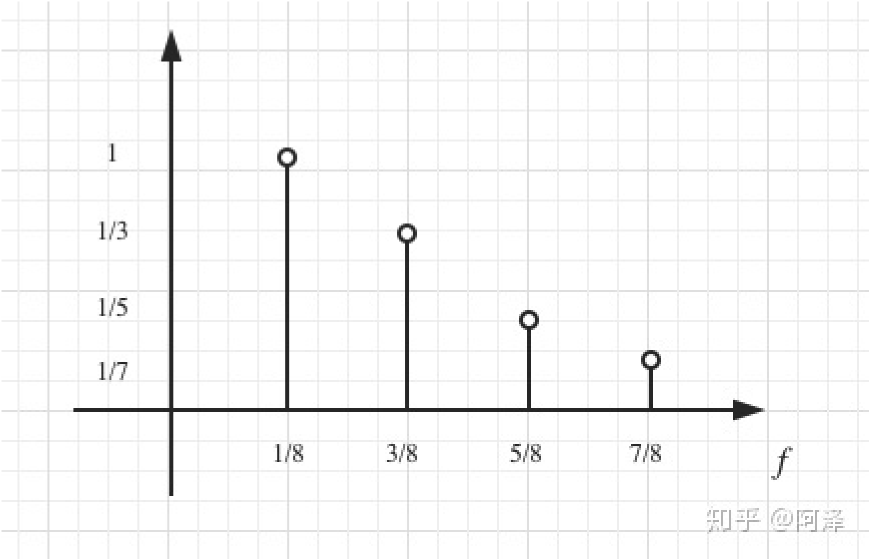


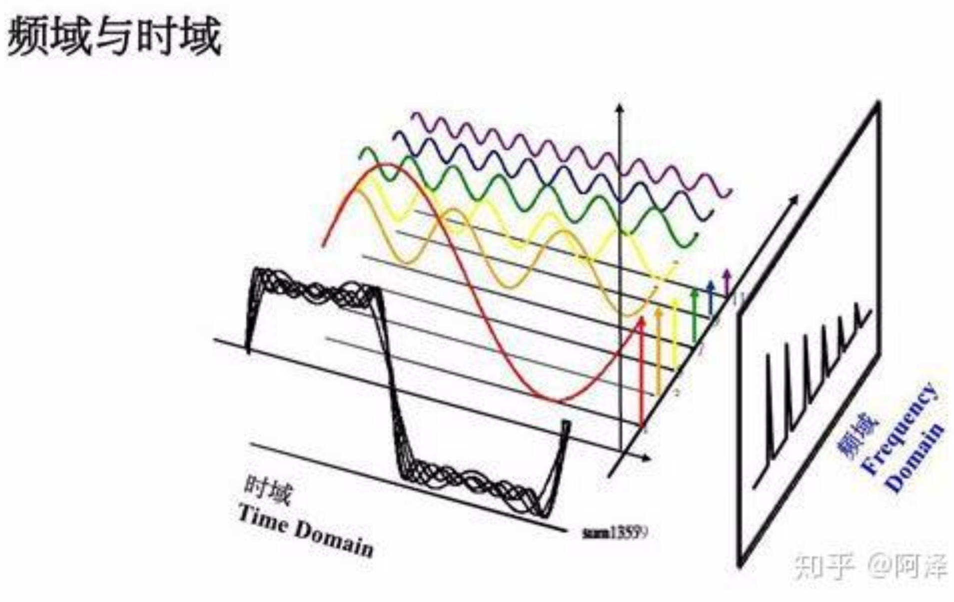
◆傅里叶变换（Fourier Transformer）



$$f(t) \approx 2.5 + \frac{10}{\pi}\left(\sin\frac{\pi t}{4} + \frac{1}{3}\sin\frac{3\pi t}{4} + \frac{1}{5}\sin\frac{5\pi t}{4} + \frac{1}{7}\sin\frac{7\pi t}{4}\right)$$

◆傅里叶变换（Fourier Transformer）





$$f(t) \approx 2.5 + \frac{10}{\pi}\left(\sin\frac{\pi t}{4} + \frac{1}{3}\sin\frac{3\pi t}{4} + \frac{1}{5}\sin\frac{5\pi t}{4} + \frac{1}{7}\sin\frac{7\pi t}{4}\right)$$

◆傅里叶变换（Fourier Transformer）

$$f(x) = \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n \sin(n\omega x + \varphi_n)$$

$$f(x) = \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n \sin(n\omega x) + \sum_{n=1}^{\infty} b_n \cos(n\omega x)$$

◆傅里叶变换（Fourier Transformer）

$$\int_0^{2\pi} cos(mx)sin(nx)dx = 0$$

$$\int_0^{2\pi} sin(mx)sin(nx)dx = \pi\delta_{mn} \quad m,n > 1$$

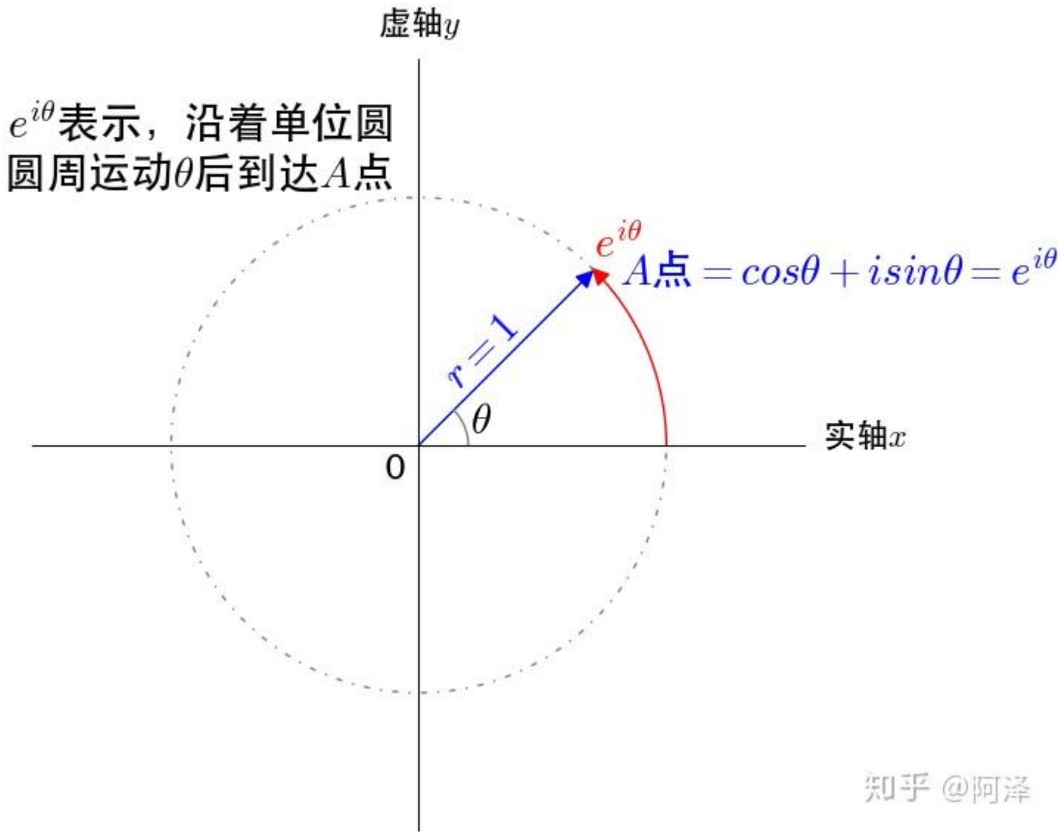
$$\int_0^{2\pi} cos(mx)cos(nx)dx = \pi\delta_{mn} \quad m,n > 1$$

$$where \quad \delta_{mn} = \begin{cases} 1 & m = n \\ 0 & m \neq n \end{cases}$$

北京理工大学 BEIJING INSTITUTE OF TECHNOLOGY

◆傅里叶变换（Fourier Transformer）

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



◆傅里叶变换（Fourier Transformer）

$$\mathcal{F}_T(\omega) = \int_{-\infty}^{+\infty} f(t) e^{-i\omega t} dt$$

◆图拉普拉斯

$$\mathbf{L} = \mathbf{D} - \mathbf{W}$$

$$\mathbf{L}_N = \mathbf{D}^{-1/2}(\mathbf{D} - \mathbf{W})\mathbf{D}^{-1/2}$$

◆图拉普拉斯

$$\Delta f = \nabla^2 f = \nabla \cdot \nabla f$$

$$\begin{aligned}
\frac{\partial^2 f}{\partial x_i^2} &= f''(x) \\
&\approx f'(x) - f'(x-1) \\
&\approx f(x+1) - f(x) - (f(x) - f(x-1)) \\
&= f(x+1) + f(x-1)) - 2f(x)
\end{aligned}$$

◆图拉普拉斯

$$
\begin{aligned}
\Delta \boldsymbol{f}_i &= \sum_{j \in N_i} \frac{\partial f_i}{\partial j^2} \\
&\approx \sum_j w_{ij}(f_i - f_j) \\
&= \sum_j w_{ij}(f_i - f_j) \\
&= (\sum_j w_{ij}) f_i - \sum_j w_{ij} f_j \\
&= d_i f_i - w_{i:} f_i
\end{aligned}
$$

$$
\begin{aligned}
\Delta f &= \begin{pmatrix} \Delta f_1 \\ \vdots \\ \Delta f_N \end{pmatrix} = \begin{pmatrix} d_1 f_1 - w_{1:} f \\ \vdots \\ d_N f_N - w_{N:} f \end{pmatrix} \\
&= \begin{pmatrix} d_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & d_N \end{pmatrix} f - \begin{pmatrix} w_{1:} \\ \vdots \\ w_{N:} \end{pmatrix} f \\
&= diag(d_i) f - \mathbf{W} f \\
&= (\mathbf{D} - \mathbf{W}) f \\
&= \mathbf{L} f
\end{aligned}
$$

◆图拉普拉斯谱分解

$$\mathbf{L} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^{-1} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^{T}$$

$$\begin{aligned} f^T\mathbf{L}f &= f^TDf - f^TWf \\ &= \sum_i d_i f_i^2 - \sum_{i,j} f_i f_j w_{ij} \\ &= \frac{1}{2}\Big(\sum_i d_i f_i^2 - 2\sum_{ij} f_i f_j w_{ij} + \sum_i d_i f_i^2\Big) \\ &= \frac{1}{2}\sum_{i,j} w_{ij}(f_i - f_j)^2 \end{aligned}$$

◆图傅里叶变换

首先考虑亥姆霍兹方程

$$\nabla^2 f = -k^2 f$$

$$\mathcal{F}_T(\omega) = \int_{-\infty}^{+\infty} f(t) e^{-i\omega t} dt$$

$$\nabla^2 e^{-i\omega t} = \frac{\partial e^{-i\omega t}}{\partial t^2} = -\omega^2 e^{-i\omega t}$$

◆图傅里叶变换

$$\mathcal{F}_T(\lambda_k) = \hat{f}_k = \sum_{i=1}^{N} f(i)u_k(i)$$

其中，$f_k = (f_k(i), \ldots, f_k(n))$ 为网络图上的 n 维向量，$f_k(i)$ 表示网络中的节点 i 的第 k 个分量，$u_k(i)$ 表示特征向量 k 的第 i 个分量。做个类比解释：特征值（频率）λ_k 下，f 的图傅立叶变换（振幅）等于 f 与 λ_x 对应的特征向量 $\mathbf{u_k}$ 的内积。

考虑矩阵乘法：

$$\hat{f} = \begin{pmatrix} \hat{f}_1 \\ \vdots \\ \hat{f}_N \end{pmatrix} = \begin{pmatrix} u_1(1) & \cdots & u_1(n) \\ \vdots & \ddots & \vdots \\ u_n(1) & \cdots & u_n(n) \end{pmatrix} \begin{pmatrix} f_1 \\ \vdots \\ f_N \end{pmatrix} = \mathbf{U}^T f$$

◆Graph Convolutional Network

$$
\begin{aligned}
(f * h)_G &= \mathcal{F}^{-1}[\mathcal{F}\{f\} \cdot \mathcal{F}\{h\}] \\
&= \mathcal{F}^{-1}[\mathbf{U}^T f \cdot \hat{h}]
\end{aligned}
$$

其中，向量 $\hat{f}$ 与向量 $\hat{h}$ 的元素点积，等价于将 $\hat{h}$ 组织成对角矩阵的形式进行矩阵乘法，所以我们有：

$$
\begin{aligned}
(f * h)_G &= \mathcal{F}^{-1}[\mathbf{U}^T f \cdot \hat{h}] \\
&= \mathcal{F}^{-1}[diag[\hat{h}_1, \ldots, \hat{h}_n]\mathbf{U}^T f]
\end{aligned}
$$

最后我们再左乘 $\mathbf{U}$ 进行逆变换：

$$
(f * h)_G = \mathbf{U} diag[\hat{h}_1, \ldots, \hat{h}_n]\mathbf{U}^T f
$$

◆GCN-1

$$y = \sigma(\mathbf{U} g_\theta \mathbf{U}^T x) = \sigma(\mathbf{U} diag[\theta_1, \ldots, \theta_n] \mathbf{U}^T x)$$

$$x_{k+1,j} = h\left(V \sum_{i=1}^{f_{k-1}} F_{k,i,j} V^T x_{k,i}\right)$$

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

◆GCN-2 ChbeyNet

$$\mathcal{F}_T(\lambda_k) = \hat{g}_k = \sum_{i=1}^{N} g(i) u_k(i)$$

$$y = \sigma(\mathbf{U} g_\theta \mathbf{U}^T x) = \sigma(\mathbf{U} g_\theta(\Lambda) \mathbf{U}^T x)$$

$$g_\theta(\Lambda) \approx \sum_{k=0}^{K-1} \theta_k \Lambda^k$$

◆GCN-2 ChbeyNet

$$y = \sigma(\mathbf{U} g_\theta(\Lambda)\mathbf{U}^T x) = \sigma(\mathbf{U}\sum_{k=0}^{K-1}\theta_k \Lambda^k \mathbf{U} x) = \sigma(\sum_{k=0}^{K-1}\theta_k L^k x)$$

设 $T_k(x)$ 为切比雪夫多项式的第 k 阶式子，切比雪夫多项式的递归式为：
$T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x)$，$T_0(x) = 1$，$T_1(x) = x$。所以我们有：

$$g_\theta(\Lambda) \approx \sum_{k=0}^{K-1}\theta_k T_k(\tilde{\Lambda})$$

其中，$\tilde{\Lambda} = \frac{2}{\lambda_{max}}\Lambda - I_N$ ；λ_{max} 是指拉普拉斯矩阵 L 的最大值。

◆GCN-3

$$g_\theta * x \approx \sum_{k=0}^{K-1} \theta_k T_k(\widetilde{L})x$$

$$g_\theta * x \approx \theta_0 x + \theta_1 (L - I_N)x = \theta_0 x - \theta_1 D^{-\frac{1}{2}} A D^{-\frac{1}{2}} x$$

$$g_\theta * x \approx \theta(I_N + D^{-\frac{1}{2}} A D^{-\frac{1}{2}})x$$

$$I_N + D^{-\frac{1}{2}} A D^{-\frac{1}{2}} \rightarrow \widetilde{D}^{-\frac{1}{2}} \tilde{A} \widetilde{D}^{-\frac{1}{2}} \quad where\ \widetilde{D}_{ii} = \sum_j \tilde{A}_{ij}\ \tilde{A} = A + I_N$$

◆GCN-3

$$H^{(l+1)}=\sigma(\widetilde{D}^{-\frac{1}{2}}\widetilde{A}\widetilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)})$$

其中，$\widetilde{A}=A+I_N$ ，A 为邻接矩阵，I_N 为单位矩阵，所以 $\widetilde{A}$ 为添加自连接的邻接矩阵；$\widetilde{D}_{ii}=\sum_j \widetilde{A}_{ij}$ ，$\widetilde{D}$ 为节点的度数矩阵；$W^{(l)}$ 为神经网络第 l 层的权重矩阵；$\sigma(\cdot)$ 是激活函数；$H^{(l)}\in R^{N\times D}$ 是第 l 层的激活矩阵，并且 $H^{(0)}=X$ ，X 是由节点 x_i 的特征向量组成矩阵。

学术分享

预告：Winograd算法（加速卷积运算）

北京理工大学宇航学院

李嘉鑫