# Deep Learning

## Episodio 7: Redes Neuronales en Grafos, Parte 1

Fernando Gama

Escuela de Graduados en Ingeniería Informática y Sistemas, Facultad de Ingeniería, UBA

18 de Agosto de 2022

# Escenas del capítulo anterior...

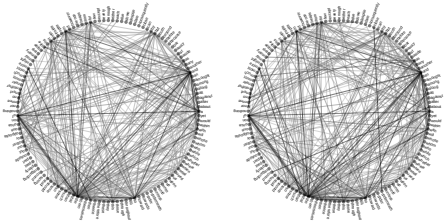
- Queremos procesar secuencias de datos
  - $\Rightarrow$ Adaptamos la red neuronal para generalizar en datos secuenciales
  - $\Rightarrow$ Compartir parámetros $\Rightarrow$ Mismos valores durante toda la secuencia

- Redes neuronales recurrentes $\Rightarrow$ Aprenden un estado oculto
  - $\Rightarrow$ Este estado es capaz de capturar la información temporal relevante

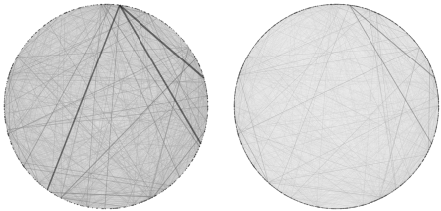
- Área activa de investigación $\Rightarrow$ Muchas extensiones
  - $\Rightarrow$ Redes neuronales bidireccionales $\Rightarrow$ El contexto importa
  - $\Rightarrow$ LSTMs, GRUs $\Rightarrow$ Poder aprender dependencias de largo alcance
- *Attention* y *Transformers* $\Rightarrow$ Aprender relaciones entre todos los elementos de la (sub)secuencia

- Los grafos son modelos para la estructura de los datos y facilitan el aprendizaje en muchas situaciones

Atribución de Autoría



Sistemas de Recomendación

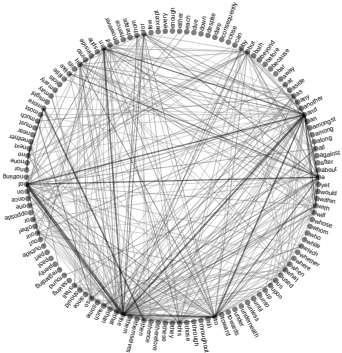


- En ambos casos existe un grafo que contiene información relevante acerca del problema a resolver

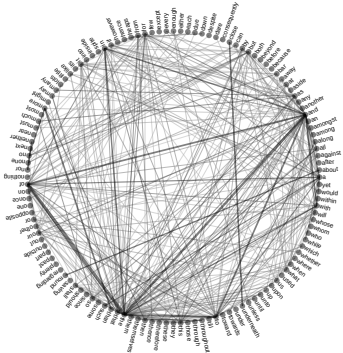
# Atribución de Autoría

- Los nodos representan distintas palabras y aristas qué tan seguido dos palabras aparecen juntas
  $\Rightarrow$ Representa las distintas formas en que los autores usan el lenguaje

William Shakespeare



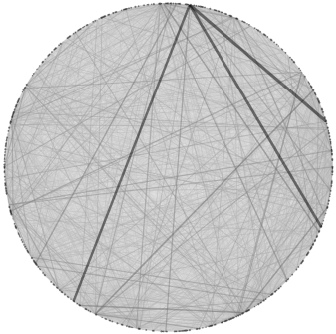
Christopher Marlowe



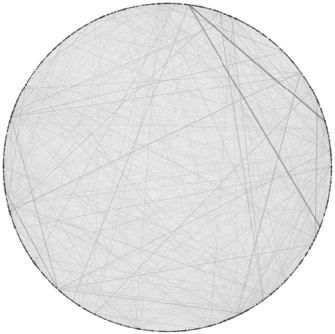
- Las diferencias en los grafos reflejan diferencias en los estilos de Shakespeare y Marlowe

- Los nodos representan distintos productos y las aristas la similitud entre las puntuaciones
  ⇒ El grafo informa en la estimación del puntaje para productos no vistos

Puntajes de un cliente



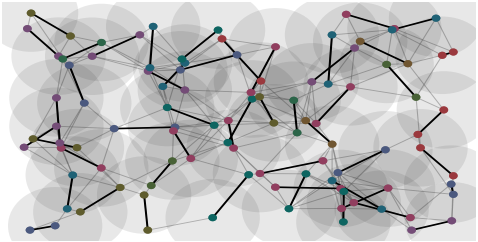
Puntajes de otro cliente



- Los grafos son más que sólo estructuras de los datos $\Rightarrow$ Sirven para modelar infraestructura de redes

Control Decentralizado de Sistemas Autónomos

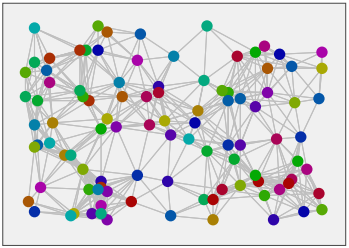
Redes de Comunicación Inalámbrica



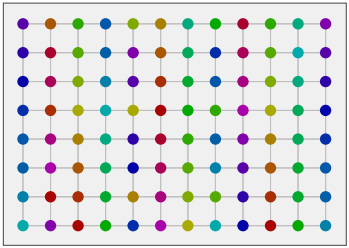
- El grafo impone una estructura de información local $\Rightarrow$ Pero queremos resolver un problema global

- Hasta ahora discutimos el *por qué* de pensar en grafos $\Rightarrow$ Pero *cómo* vamos a procesarlos?
- Las justificaciones teóricas y la inmensa evidencia empírica sugieren elegir una red neuronal

Queremos una red neuronal sobre esto



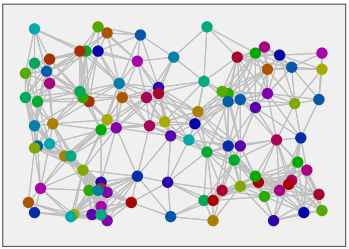
Pero sabemos usar redes neuronales acá



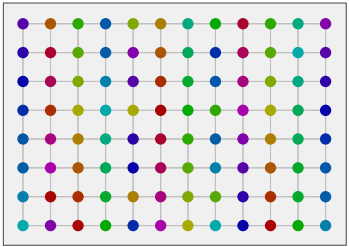
- Las redes neuronales completas (MLP) no escalan a grandes dimensiones $\Rightarrow$ Pero las CNNs sí

- Las CNNs son una cascada de capas $\Rightarrow$ Convolución seguido de activación no-lineal

Usamos una red neuronal en el grafo



Usamos una red neuronal convolucional



- Generalizar la operación de convolución a grafos y luego aplicar activación no-lineal
- Poner capas en cascada para crear una red neuronal en el grafo (GNN)

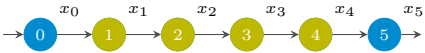
- ¿Cómo generalizamos las convoluciones para ser capaces de operar en grafos?
  $\Rightarrow$ Podemos describir la estructura temporal usando grafos que soportan señales en tiempo discreto

Descripción del tiempo con un grafo de línea dirigido


- El grafo de línea dirigido representa la adyacencia de los puntos en el tiempo

- Las convoluciones son combinaciones lineales de desplazamientos de la señal

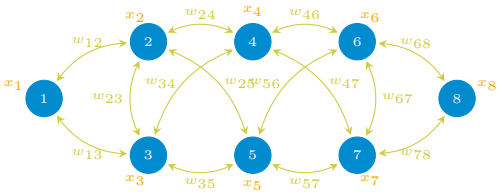
Descripción del tiempo con un grafo de línea dirigido



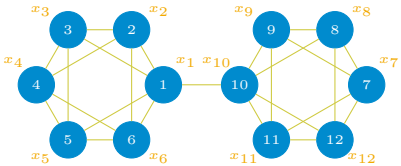

- Filtro con coeficientes $h_k$ $\Rightarrow$ Salida $\mathbf{z} = h_0\mathsf{D}_0(\mathbf{x}) + h_1\mathsf{D}_1(\mathbf{x}) + h_2\mathsf{D}_2(\mathbf{x}) + h_3\mathsf{D}_3(\mathbf{x}) + \ldots = \sum_{k=0}^{\infty} h_k\,\mathsf{D}_k(\mathbf{x})$

- Las señales temporales (y las imágenes) son importantes, pero son una clase limitada de señales
- Usamos grafos como descripciones genéricas de estructuras de la señal
  $\Rightarrow$ Con valores de la señal asociados a los nodos y aristas que determinan similitud

Una señal sobre un grafo



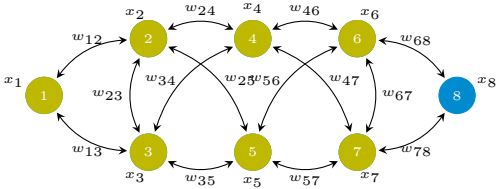

Otra señal sobre otro grafo



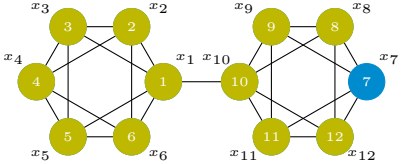

- Nodos, señales y aristas. Productos, puntajes y correlaciones. Drones, velocidades y distancias.

- Para señales en grafos definimos la convolución como una combinación lineal de desplazamientos

Una señal sobre un grafo




Otra señal sobre otro grafo

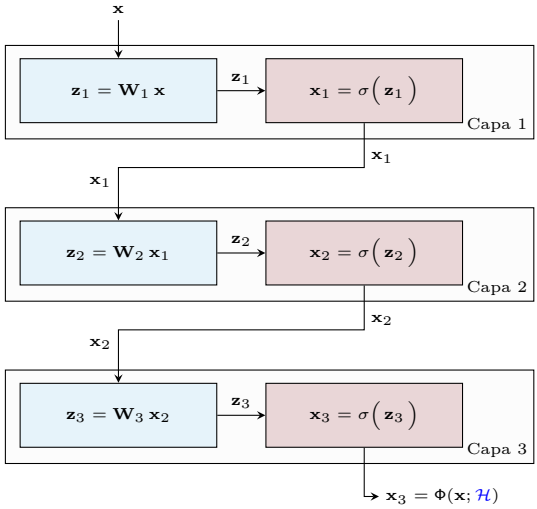



- Filtro con coeficientes $h_k$ $\Rightarrow$ Salida $\mathbf{z} = h_0\mathbf{D}_0(\mathbf{x}) + h_1\mathbf{D}_1(\mathbf{x}) + h_2\mathbf{D}_2(\mathbf{x}) + h_3\mathbf{D}_3(\mathbf{x}) + \ldots = \sum_{k=0}^{\infty} h_k\mathbf{D}_k(\mathbf{x})$
- Cómo definimos el desplazamiento para que relacione elementos contiguos de la señal

# Redes Neuronales (NNs)

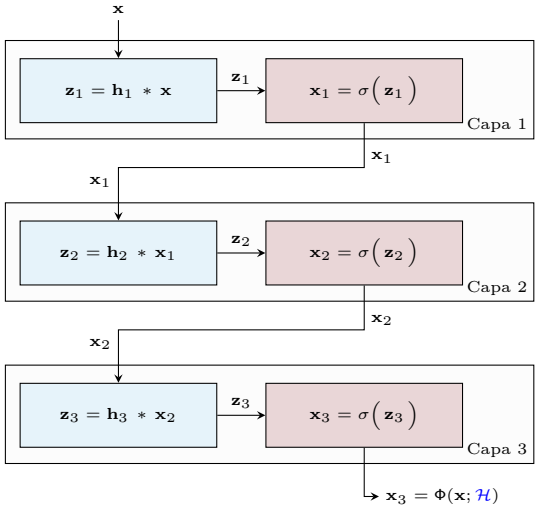
- Una red neuronal es una cascada de bloques o capas
- Cada una aplica una transformación lineal

  $\Rightarrow$ Y una función de activación no-lineal
- No escala para datos $\mathbf{x}$ de gran dimensión



# Redes Neuronales Convolucionales (CNNs)

- Una NN convolucional es una cascada de capas
- Cada una aplica una convolución
  ⇒ Y una función de activación no-lineal
- Escala a datos grandes sin problemas
- Una CNNs es una variación de un filtro convolucional
  ⇒ Le agregamos una activación no-lineal y repetimos
  ⇒ Escalan porque las convoluciones escalan



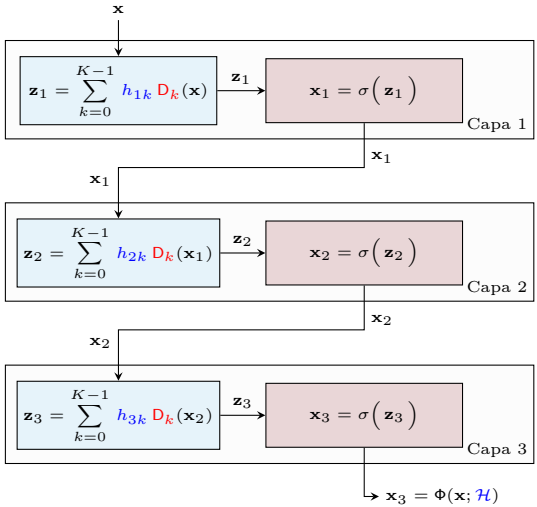
# Pensando Señales Temporales como un Grafo

- Las convoluciones son combinaciones lineales

  $\Rightarrow$ De desplazamientos determinados por este grafo

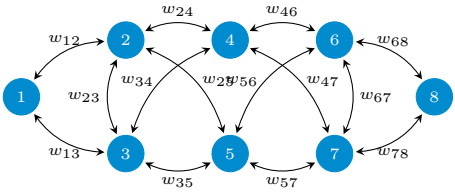

- Esta una manera de escribir la convolución

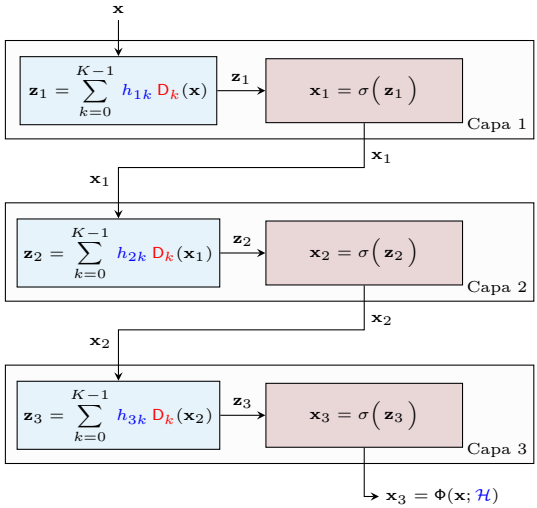
  $\Rightarrow$ Que permite generalizar a otros grafos



# Redes Neuronales en Grafos (GNNs)

- El grafo puede ser cualquier grafo
- Hay que definir la noción de desplazamiento

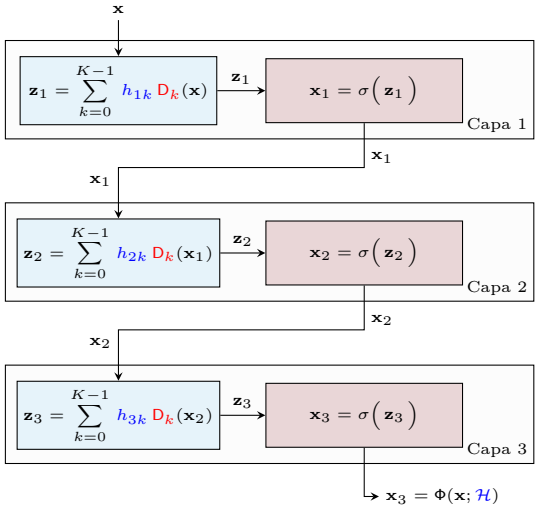


# Redes Neuronales en Grafos (GNNs)

- Una NN en grafos es una cascada de capas
- Cada capa aplica una convolución en el grafo

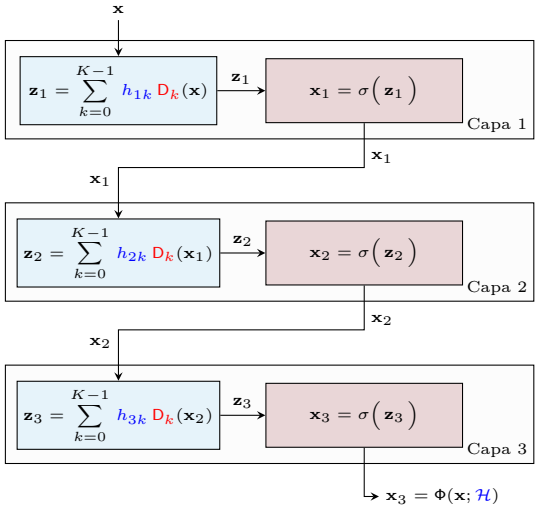
  $\Rightarrow$ Y una función de activación no-lineal
- Es un MLP con la operación lineal restringida
- Recupera la CNN si es un grafo de línea dirigido



# Redes Neuronales en Grafos (GNNs)

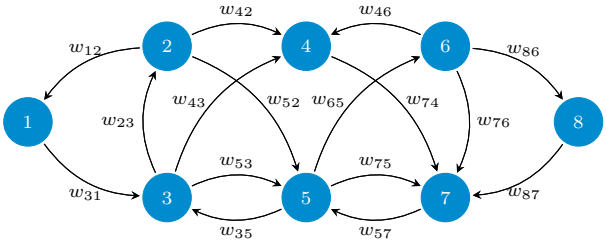
- Hay evidencia empírica de que las GNNs escalan
- Una GNN es una variación de un filtro en un grafo

  $\Rightarrow$ Se le agrega una activación no-lineal y se repite
- Escala porque el filtro explota la estructura

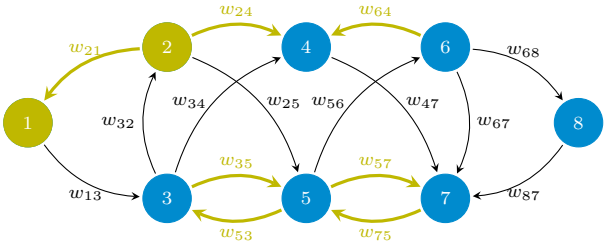


# Nodos, Aristas y Pesos

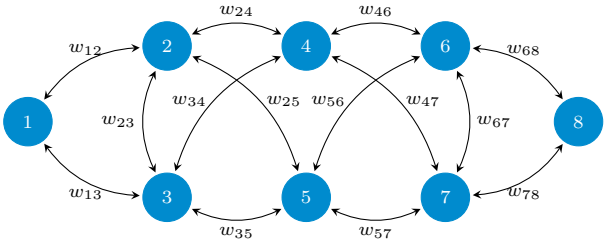
- Un grafo es una tríada $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathsf{W})$ de nodos, aristas y pesos
  - $\Rightarrow$ Los nodos son un conjunto de $N$ elementos (e.g., $\mathcal{V} = \{1, \ldots, N\}$ o $\mathcal{V} = \{v_1, \ldots, v_N\}$)
  - $\Rightarrow$ Las aristas son pares ordenados de nodos $(i, j)$ (determinan relaciones de influencia)
  - $\Rightarrow$ Los pesos $w_{ij} \in \mathbb{R}$ son números asociados a las aristas $(i, j)$ (la magnitud de la influencia)



- La arista $(i, j)$ se representa como una flecha apuntando de $i$ a $j$ $\Rightarrow$ Influencia de $i$ a $j$
- La arista $(i, j)$ es diferente de la arista $(j, i)$ $\Rightarrow$ Es posible que $(i, j) \in \mathcal{E}$ y $(j, i) \notin \mathcal{E}$ simultáneamente
- Si las dos aristas existen, puede ser que sus pesos sean distintos $\Rightarrow$ Es posible que $w_{ij} \neq w_{ji}$

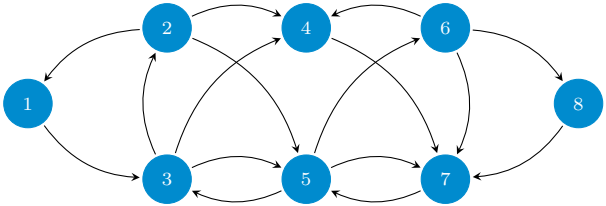


- Un grafo es no dirigido si tanto su conjunto de aristas como su función de pesos son simétricas
  - $\Rightarrow$ Las aristas vienen de a pares $\Rightarrow$ Tenemos $(i, j) \in \mathcal{E}$ si y sólo si $(j, i) \in \mathcal{E}$
  - $\Rightarrow$ Los pesos son simétricos $\Rightarrow$ $\mathsf{W}$ cumple que $\mathsf{W}(i, j) = w_{ij} = w_{ji} = \mathsf{W}(j, i)$ para todo $(i, j) \in \mathcal{E}$



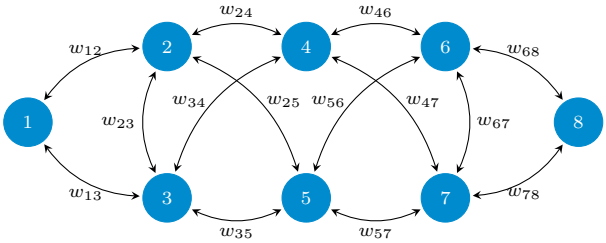
# Grafos sin Peso

- Un grafo es sin peso si no tiene una función de peso W
  - $\Rightarrow$ En general, se asume entonces, que todos los pesos son unitarios $\Rightarrow w_{ij} = 1$ para todo $(i, j) \in \mathcal{E}$
- Los grafos sin peso pueden ser dirigidos o no dirigidos



# Grafos No Dirigidos con Peso

- Los grafos pueden ser dirigidos o no dirigidos, pueden tener peso o no tenerlo
- En general, vamos a considerar grafos que sean no dirigidos y con pesos

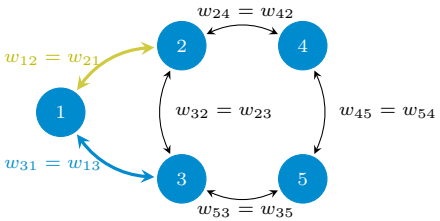


# Matrices de Adyacencia

- La matriz de adyacencia del grafo $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathsf{W})$ es una matriz rala $\mathbf{A} \in \mathbb{R}^{N \times N}$ con elementos no-nulos

$$[\mathbf{A}]_{ij} = w_{ji}, \text{ para todo } (j, i) \in \mathcal{E}$$

- Si el grafo es no dirigido, la matriz de adyacencia es simétrica $\Rightarrow \mathbf{A} = \mathbf{A}^\mathsf{T}$ (como en el ejemplo)

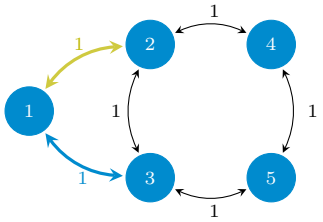



$$\mathbf{A} = \begin{bmatrix} 0 & w_{21} & w_{31} & 0 & 0 \\ w_{12} & 0 & w_{32} & w_{42} & 0 \\ w_{13} & w_{23} & 0 & 0 & w_{53} \\ 0 & w_{24} & 0 & 0 & w_{54} \\ 0 & 0 & w_{35} & w_{45} & 0 \end{bmatrix}.$$

# Matrices de Adyacencia

- Para el caso particular en que el grafo no tiene pesos $\Rightarrow$ La matriz de adyacencia tiene pesos unitarios

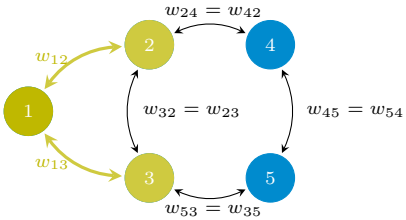
$$[\mathbf{A}]_{ij} = 1, \text{ for all } (j,i) \in \mathcal{E}$$




$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix}.$$

# Vecindario y Grado

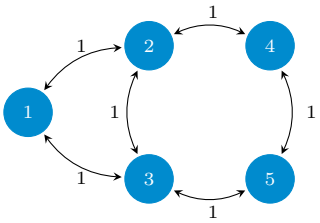
- El vecindario del nodo $i$ es el conjunto de nodos que influencian a $i$ $\Rightarrow n(i) := \{j : (j,i) \in \mathcal{E}\}$
- El grado $d_i$ del nodo $i$ es la suma de los pesos de las aristas incidentes $\Rightarrow d_i = \sum_{j \in n(i)} w_{ij} = \sum_{\{j:(j,i)\in\mathcal{E}\}} w_{ij}$




- Vecindario del nodo 1 $\Rightarrow n(1) = \{2, 3\}$
- Grado del nodo 1 $\Rightarrow d_1 = w_{21} + w_{31}$

# Matriz de Grado

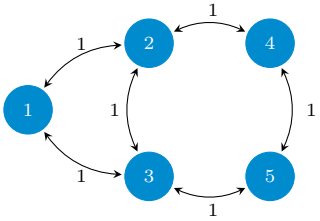
- La matriz de grado es una matriz diagonal $\mathbf{D}$ con los grados en la diagonal $\Rightarrow [\mathbf{D}]_{ii} = d_i$
- Se puede escribir en función de la matriz de adyacencia $\mathbf{D} = \mathsf{diag}(\mathbf{A1})$




$$\mathbf{D} = \begin{bmatrix} 2 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 2 \end{bmatrix}.$$

# Matriz Laplaciana

- La matriz Laplaciana dada un a matriz de adyacencia $\mathbf{A}$ es $\Rightarrow \mathbf{L} = \mathbf{D} - \mathbf{A} = \mathsf{diag}(\mathbf{A1}) - \mathbf{A}$
- Se puede escribir directamente usando los pesos del grafo $[\mathbf{A}]_{ij} = w_{ji}$ (sólo para grafos no dirigidos)
  - $\Rightarrow$ Los elementos fuera de la digonal $\Rightarrow [\mathbf{L}]_{ij} = -[\mathbf{A}]_{ij} = -w_{ji}$
  - $\Rightarrow$ Los elementos en la diagonal $\Rightarrow [\mathbf{L}]_{ii} = d_i = \sum_{j \in n(i)} w_{ji}$




$$\mathbf{L} = \begin{bmatrix} 2 & -1 & -1 & 0 & 0 \\ -1 & 3 & -1 & -1 & 0 \\ -1 & -1 & 3 & 0 & -1 \\ 0 & -1 & 0 & 2 & -1 \\ 0 & 0 & -1 & -1 & 2 \end{bmatrix}.$$

# Normalizaciones

- Las normalizaciones de la matriz de adyacencia y Laplaciana expresan los pesos relativos al grado
- La matriz de adyacencia normalizada $\Rightarrow \bar{\mathbf{A}} := \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$ (es simétrica si el grafo es no dirigido)
- La matriz Laplaciana normalizada $\Rightarrow \bar{\mathbf{L}} := \mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2}$ (sólo existe en grafos no dirigidos)
- Dadas las definiciones de las normalizaciones de las matrices de adyacencia y Laplaciana $\Rightarrow \bar{\mathbf{L}} := \mathbf{I} - \bar{\mathbf{A}}$

# Operador de Desplazamientos en el Grafo

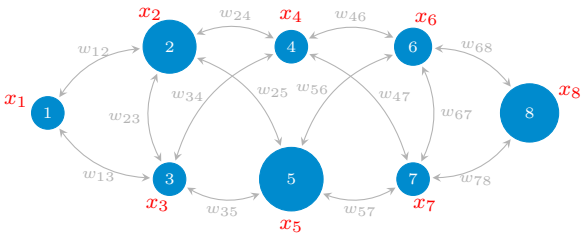
- El operador desplazamiento en el grafo $\mathbf{S}$ es un genérico para cualquier matriz del grafo

| Matriz de adyacencia | Matriz Laplaciana | Adyacencia normalizada | Laplaciana normalizada |
|---|---|---|---|
| $\mathbf{S} = \mathbf{A}$ | $\mathbf{S} = \mathbf{L}$ | $\mathbf{S} = \bar{\mathbf{A}}$ | $\mathbf{S} = \bar{\mathbf{L}}$ |

- Si el grafo es no dirigido el operador desplazamiento $\mathbf{S}$ es simétrico $\Rightarrow \mathbf{S} = \mathbf{S}^{\mathsf{T}}$
- La elección específica de ODG importa en la práctica pero la los análisis suelen valer para cualquier $\mathbf{S}$

# Señales en Grafos

- Consideremos un grafo $\mathcal{G}$ dado con $N$ nodos y operador deplazamiento $\mathbf{S}$
- Una señal en el grafo es un vector $\mathbf{x} \in \mathbb{R}^N$ tal que el elemento $x_i$ se asocia al nodo $i$
- Para no perder de vista que la estructura de la señal depende del grafo $\Rightarrow$ La señal es el par $(\mathbf{S}, \mathbf{x})$

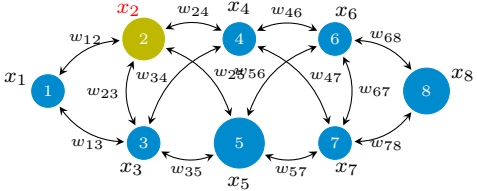


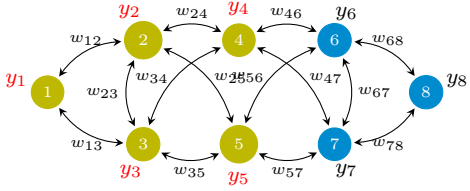

- El grafo es una suposición de similaridad o proximidad entre los elementos de $\mathbf{x}$ (contiguos o no)

# Difusión de Señales en el Grafo

- Multiplicar la señal $\mathbf{x}$ con el operador desplazamiento $\mathbf{S}$ difunde la señal
- La señal difundida $\mathbf{y} = \mathbf{S}\mathbf{x}$ tiene elementos dados por $y_i = \sum_{j \in n(i)} w_{ji} x_j$

  $\Rightarrow$ Captura una operación local donde los elementos se mezclan con los de los vecinos



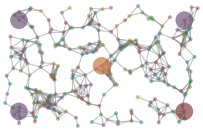



# La Secuencia de Difusión

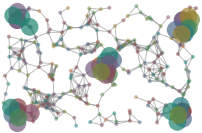
- El ODG se puede repetir para producir la secuencia de difusión definida recursivamente como

$$\mathbf{x}^{(k+1)} = \mathbf{S}\mathbf{x}^{(k)}, \qquad \text{con} \quad \mathbf{x}^{(0)} = \mathbf{x}$$

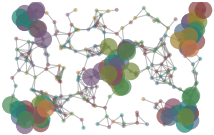
- Alternativamente, desdoblamos la recursión para escribir $\mathbf{x}^{(k)} = \mathbf{S}^k\mathbf{x}$



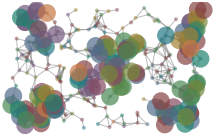
$\mathbf{x}^{(0)} = \mathbf{x} = \mathbf{S}^0\mathbf{x}$



$\mathbf{x}^{(1)} = \mathbf{S}\mathbf{x}^{(0)} = \mathbf{S}^1\mathbf{x}$



$\mathbf{x}^{(2)} = \mathbf{S}\mathbf{x}^{(1)} = \mathbf{S}^2\mathbf{x}$



$\mathbf{x}^{(3)} = \mathbf{S}\mathbf{x}^{(2)} = \mathbf{S}^3\mathbf{x}$

- El $k$-ésimo elemento de la secuencia $\mathbf{x}^{(k)}$ difunde la información al vecindario a $k$-saltos

- Dado un ODG $\mathbf{S}$ y coeficientes coefficients $h_k$, un filtro en un grafo es un polinomio en $\mathbf{S}$

$$\mathbf{H}(\mathbf{S}) = \sum_{k=0}^{\infty} h_k \mathbf{S}^k$$

- Aplicar un filtro $\mathbf{H}(\mathbf{S})$ a una señal $\mathbf{x}$ resulta en otra señal

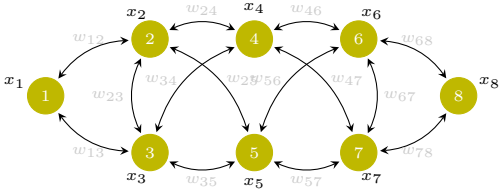
$$\mathbf{y} = \mathbf{H}(\mathbf{S})\mathbf{x} = \sum_{k=0}^{\infty} h_k \mathbf{S}^k \mathbf{x}$$

- Decimos que $\mathbf{y} = \mathbf{h} *_{\mathbf{S}} \mathbf{x}$ es la convolución en el grafo del filtro $\mathbf{h} = \{h_k\}_{k=0}^{\infty}$ con la señal $\mathbf{x}$

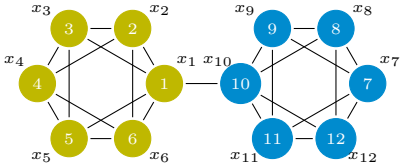
# De Información Local a Información Global

- Las convoluciones en grafos agregan información creciente desde lo local hasta lo global
  $\Rightarrow$ Conceptualmente, hacen lo mismo que las convoluciones en el tiempo

Filtro en un grafo



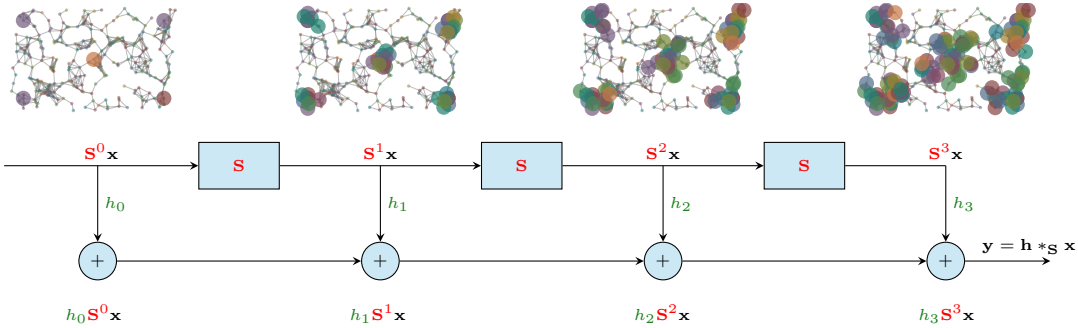

El mismo filtro en otro grafo




- Filtro con coeficientes $\mathbf{h} = \{h_k\}_{k=0}^{\infty}$ $\Rightarrow \mathbf{z} = h_0\mathbf{S}^0\mathbf{x} + h_1\mathbf{S}^1\mathbf{x} + h_2\mathbf{S}^2\mathbf{x} + h_3\mathbf{S}^3\mathbf{x} + \ldots = \sum_{k=0}^{\infty} h_k\mathbf{S}^k\mathbf{x}$
- La salida del filtro en el grafo depende de los coeficientes $\mathbf{h}$ y del operador desplazamiento $\mathbf{S}$

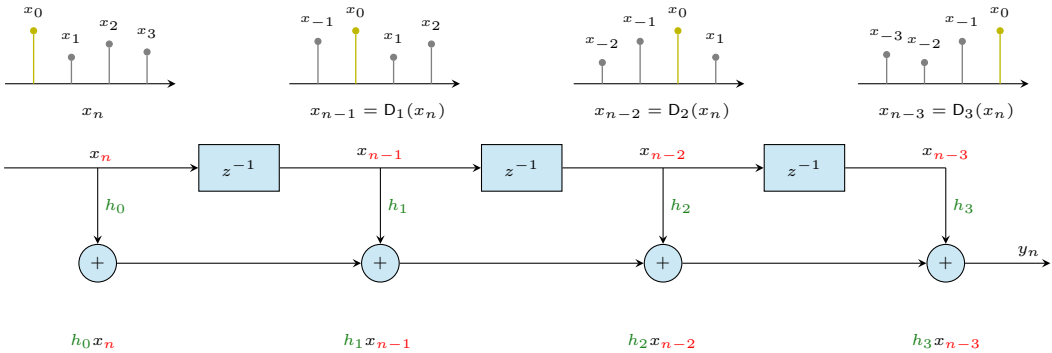
# Filtros Convolucionales en el Grafo como Difusiones

- Una convolución en el grafo es una combinación lineal de los elementos de la secuencia de difusión
- Podemos representar la convolución en el grafo como un diagrama en bloques



# Convoluciones en el Tiempo

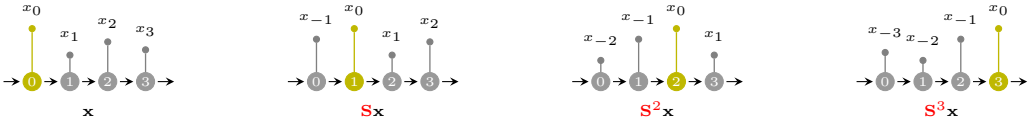
- Los filtros convolucionales procesan señales temporales explotando el desplazamiento




- La convolución temporal es una combinación lineal de desplazamiento $\Rightarrow y_n = \sum_{k=0}^{K-1} h_k x_{n-k}$

# Señales Temporales como Señales en el Grafo

- Las señales temporales se pueden pensar como señales en grafos sobre un grafo de línea $\mathbf{S}$



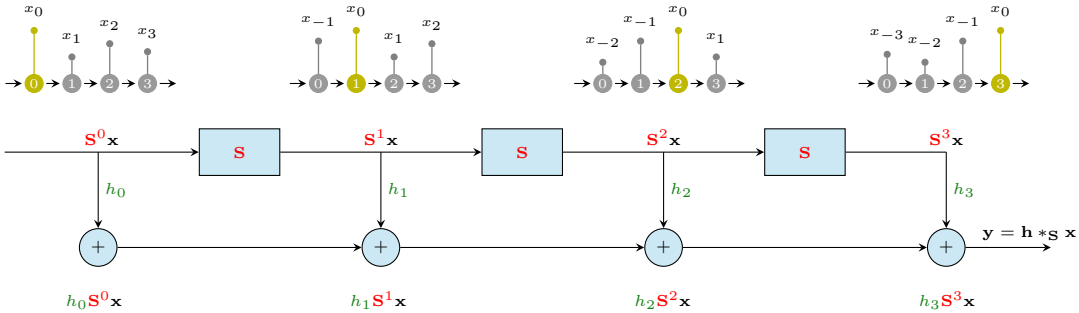

- El desplazamiento temporal es equivalente a aplicar el ODG $\mathbf{S}$ del grafo de línea

$$\mathbf{S}^3\mathbf{x} = \mathbf{S}(\mathbf{S}^2\mathbf{x}) = \begin{bmatrix} & \vdots & \vdots & \vdots & \\ \cdots & 0 & 0 & 0 & \cdots \\ \cdots & 1 & 0 & 0 & \cdots \\ \cdots & 0 & 1 & 0 & \cdots \\ \cdots & 0 & 0 & 1 & \cdots \\ & \vdots & \vdots & \vdots & \end{bmatrix} \begin{bmatrix} \vdots \\ x_{-2} \\ x_1 \\ x_0 \\ x_1 \\ \vdots \end{bmatrix} = \begin{bmatrix} \vdots \\ x_{-3} \\ x_{-2} \\ x_{-1} \\ x_0 \\ \vdots \end{bmatrix}$$

- Los desplazamientos son potencias del operador desplazamiento aplicado a la señal
  $\Rightarrow$ Por lo tanto, los filtros convolucionales son polinomios en $\mathbf{S}$ $\Rightarrow \mathsf{D}_k(\mathbf{x}) = \mathbf{S}^k\mathbf{x}$

# La Convolución como un Polinomio

- La convolución es una combinación lineal de desplazamientos de la señal
- Los desplazamientos temporales se obtienen multiplicando con la adyacencia $\mathbf{S}$ del grafo de línea

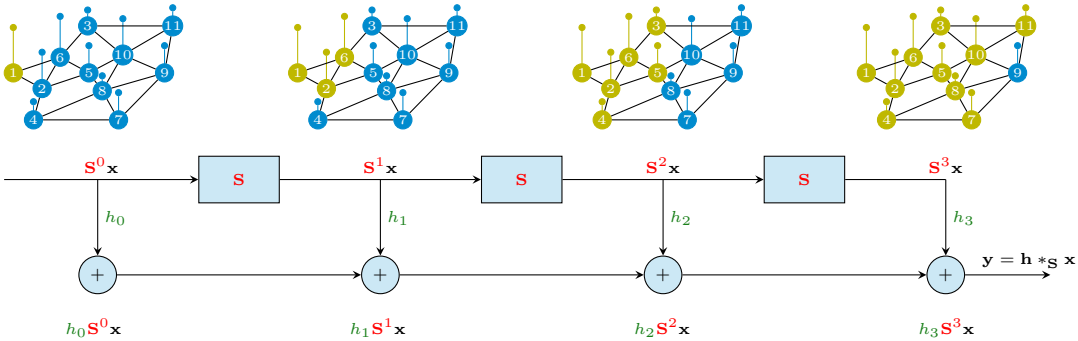



- La convolución es un polinomio en la matriz de adyacencia del grafo $\Rightarrow \mathbf{y} = \mathbf{h} * \mathbf{x} = \sum_{k=0}^{K-1} h_k \mathbf{S}^k \mathbf{x}$

# La Convolución en Grafos

- Para un grafo arbitrario con ODG $\mathbf{S}$ la convolución es una combinación lineal de desplazamientos

$$\mathsf{h} * \mathsf{x} = \sum_{k=0}^{K-1} h_k \mathsf{D}_k(\mathsf{x}) = \sum_{k=0}^{K-1} h_k \mathbf{S}^k \mathbf{x}$$



# Minimización del Riesgo Empírico

- El objetivo, recordemos, es minimizar el riesgo empírico
- En el problema de minimización del riesgo empírico, tenemos:
  - $\Rightarrow$ Un conjunto de muestras de entrenamiento $\mathcal{T} = \{\mathbf{x}\}$
  - $\Rightarrow$ Una función de pérdida $\mathsf{J}(\Phi(\mathbf{x}))$ para evaluar el algoritmo $\Phi$
  - $\Rightarrow$ Una clase de funciones $\boldsymbol{\Phi}$ de donde tomamos el algoritmo de aprendizaje $\Phi \in \boldsymbol{\Phi}$
- Encontrar un algoritmo $\Phi^\star \in \boldsymbol{\Phi}$ que minimiza la pérdida $\mathsf{J}\Big(\Phi(\mathbf{x})\Big)$ promediado sobre las muestras

$$\Phi^\star = \underset{\Phi \in \boldsymbol{\Phi}}{\arg\,\min} \sum_{\mathbf{x} \in \mathcal{T}} \mathsf{J}\Big(\Phi(\mathbf{x})\Big)$$

- Usamos $\Phi^\star(\mathbf{x})$ para estimar la salida $\hat{\mathbf{y}} = \Phi^\star(\mathbf{x})$ cuando la entrada $\mathbf{x}$ no pertenecen al entrenamiento

# Minimización del Riesgo Empírico

- En estos problemas, la clase de funciones $\mathbf{\Phi}$ es elección del diseñador

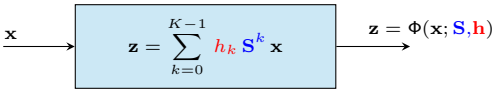
$$\mathbf{\Phi}^\star = \arg\min_{\mathbf{\Phi}\in\mathbf{\Phi}} \sum_{\mathbf{x}\in\mathcal{T}} \mathsf{J}\Big(\mathbf{\Phi}(\mathbf{x})\Big)$$

- Diseñar un algoritmo de aprendizaje automático $\equiv$ encontrar la clase de funciones $\mathbf{\Phi}$ adecuada

- Cuando trabajamos con señales en grafos, elegir filtros convolucionales en grafos parece una buena idea

# Aprendiendo con un Filtro Convolucional en el Grafo

- Asumimos que los datos $\mathbf{x}$ son señales en el grafo estructuradas mediante un grafo con ODG $\mathbf{S}$
- La clase de funciones son filtros en grafos de orden $K$ sobre el ODG $\mathbf{S}$ $\Rightarrow \boldsymbol{\Phi}(\mathbf{x};\mathbf{S},\mathbf{h}) = \sum_{k=0}^{K-1} h_k \mathbf{S}^k \mathbf{x}$

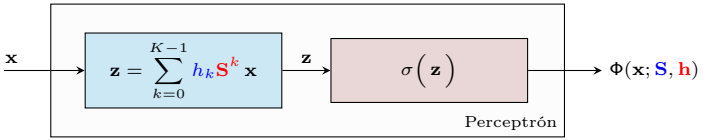



- Aprender minimizando el riesgo, restringidos a la clase de filtros $\Rightarrow \mathbf{h}^\star = \arg\min_{\mathbf{h}} \sum_{\mathbf{x}\in\mathcal{T}} \mathsf{J}\Big(\boldsymbol{\Phi}(\mathbf{x};\mathbf{S},\mathbf{h})\Big)$

  $\Rightarrow$ La optimización es sobre los coeficientes $\mathbf{h}$ para el ODG $\mathbf{S}$ dado

# Aprendiendo con un Perceptrón en el Grafo

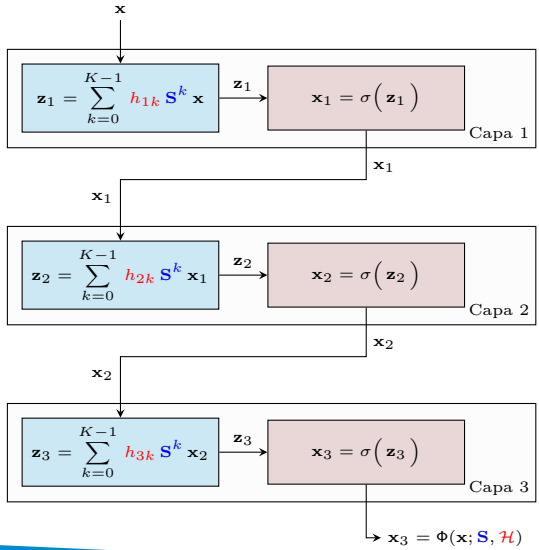
- Los filtros tienen capacidad de representación limitada $\Rightarrow$ Sólo pueden aprender funciones lineales
- Elegir $\boldsymbol{\Phi}$ como el conjunto de perceptrones en el grafo $\Rightarrow \boldsymbol{\Phi}(\mathbf{x}) = \boldsymbol{\Phi}(\mathbf{x}; \mathbf{S}, \mathbf{h}) = \sigma\left( \sum_{k=0}^{K-1} h_k \mathbf{S}^k \mathbf{x} \right)$




- El algoritmo óptimo va a ser un perceptrón $\Rightarrow \mathbf{h}^\star = \arg\min_{\mathbf{h}} \sum_{\mathbf{x} \in \mathcal{T}} \mathrm{J}\left( \boldsymbol{\Phi}(\mathbf{x}; \mathbf{S}, \mathbf{h}) \right)$

  $\Rightarrow$ El perceptrón puede aprender relaciones no-lineales $\Rightarrow$ Mayor poder de representación

- Incrementar la representación con una GNN
- Cascada de algunos perceptrones en el grafo
  - $\Rightarrow$ La señal de entrada $\mathbf{x}$ en la capa 1
  - $\Rightarrow$ La salida de la capa 1 es la entrada de la capa 2
  - $\Rightarrow$ La salida de la capa 2 es la entrada de la capa 3
- La última capa es la salida de la GNN $\Rightarrow \boldsymbol{\Phi}(\mathbf{x}; \mathbf{S}, \mathcal{H})$
  - $\Rightarrow$ Parametrizada por los coeficientes $\mathcal{H} = [\mathbf{h}_1, \mathbf{h}_2, \mathbf{h}_3]$



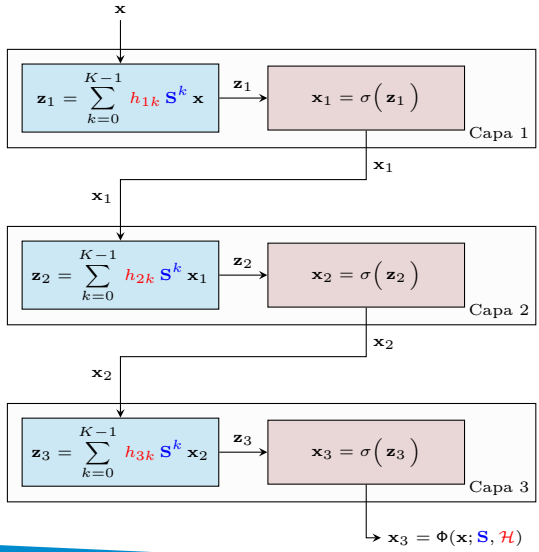
# Aprendiendo con una Red Neuronal en el Grafo

- Aprender los filtros $\mathcal{H}^\star = (\mathbf{h}_1^\star, \mathbf{h}_2^\star, \mathbf{h}_3^\star)$ de la GNN

$$\mathcal{H}^\star = \arg\min_{\mathcal{H}} \sum_{\mathbf{x}\in\mathcal{T}} \mathsf{J}\Big(\mathbf{\Phi}(\mathbf{x}; \mathbf{S}, \mathcal{H})\Big)$$

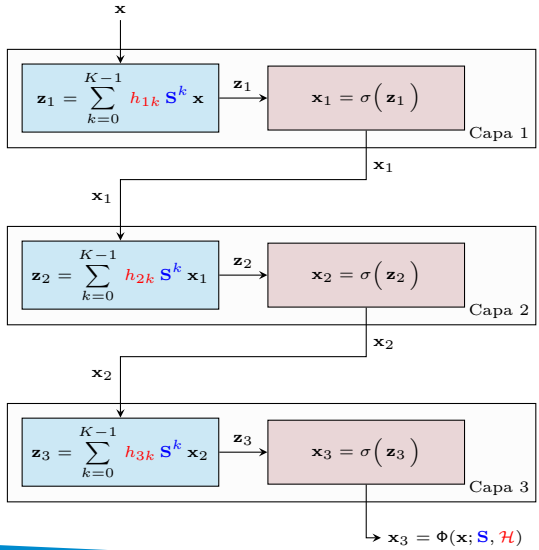
- Optimizamos sobre el filtro $\Rightarrow$ El grafo $\mathbf{S}$ viene dado

  $\Rightarrow$ Captura información que explota la GNN



# Redes Neuronales en el Grafo y Filtros en el Grafo

- Las GNNs son variaciones de los filtros en grafos
- Sólo se agrega la activación no-lineal
  - ⇒ Los nodos se procesan por separado
  - ⇒ No se mezcla información
- Las GNNs funcionan mejor que los filtros en grafos
  - ⇒ Se puede explicar mediante análisis de estabilidad



# Transferencia de GNNs a Distintos Grafos

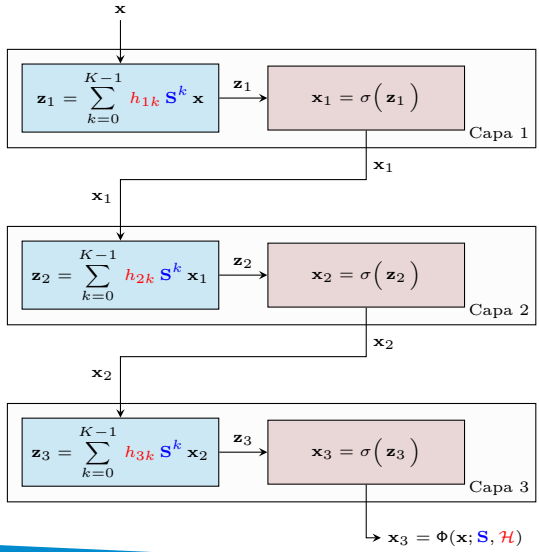
- La GNN depende del grafo $\mathbf{S}$
- Podemos pensar $\mathbf{S}$ como un parámetro

  $\Rightarrow$ Captura información sobre la estructura

- O podemos pensar que $\mathbf{S}$ es la entrada

  $\Rightarrow$ Permite transferir a diferentes grafos

  $\Rightarrow$ Se aprenden sólo los coeficientes $\mathcal{H}^{\star}$



# Resumen del Capítulo

- La importancia de los datos con estructuras descriptas por grafos
  $\Rightarrow$ Requieren tener en cuenta el grafo para procesar de manera adecuada

- Señales en grafos, operador de desplazamiento en el grafo
  $\Rightarrow$ Convolución en grafos $\Rightarrow$ Combinación lineal de señales en el vecindario

- Redes neuronales en grafos $\Rightarrow$ Reemplazar la operación lineal por una convolución en el grafo
  $\Rightarrow$ Explota la estructura de los datos descripta por el grafo

# Escenas del Próximo Capítulo...

- Extensión de GNNs a señales en grafos con múltiples features
  $\Rightarrow$ Se utiliza un banco de filteros para la convolución en el grafo
- Pooling respetando al estructura del grafo $\Rightarrow$ Resumen local, selección de nodos
- Sistemas de recomendación $\Rightarrow$ Las similitudes de los puntajes son un grafo
- Atribución de autoría $\Rightarrow$ Las palabras son nodos, y la co-ocurrencia son aristas
  $\Rightarrow$ Atribuir textos cortos $\Rightarrow$ Histogramas de palabras en textos cortos
- Controlar enjambres de robots $\Rightarrow$ Aprendizaje por imitación, transferencia, escalabilidad