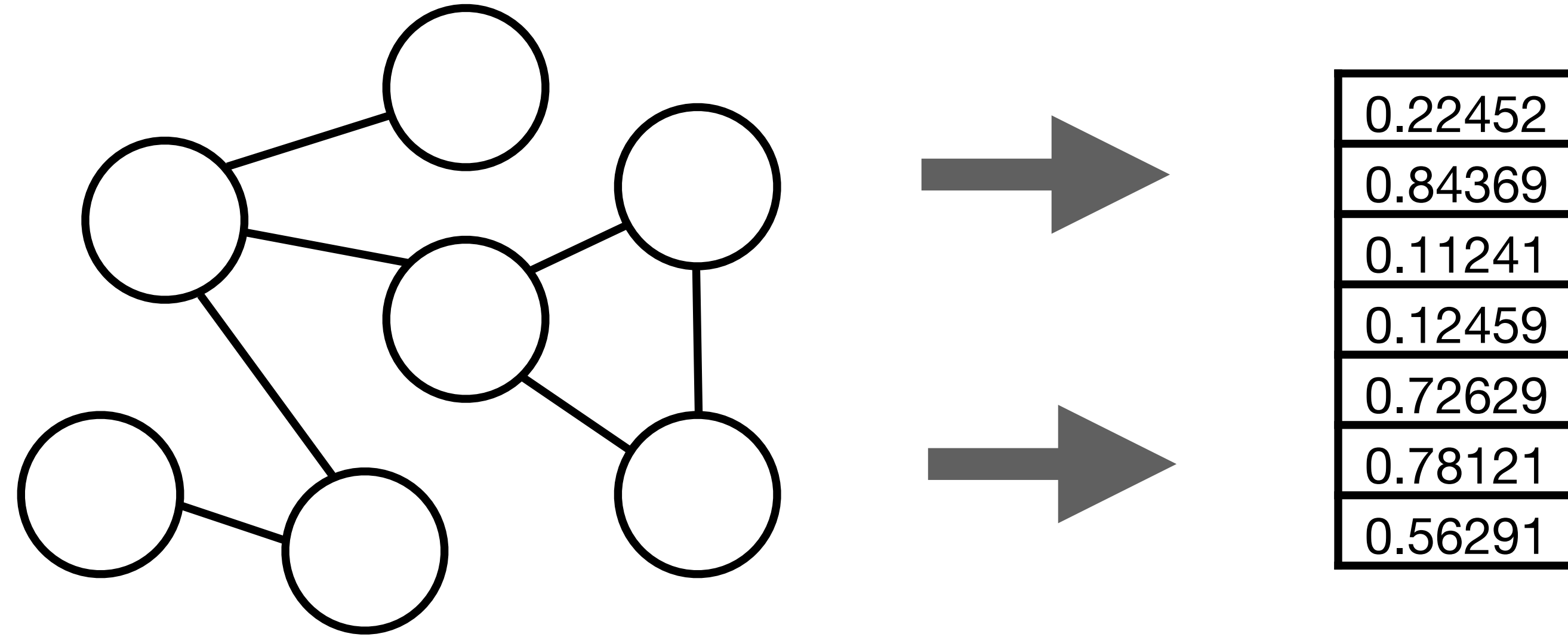


Graph Neural Networks en la práctica

Graph Neural Networks

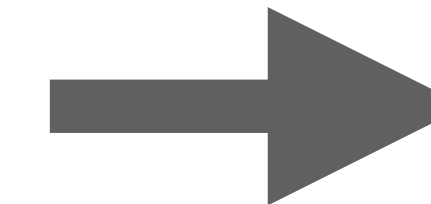
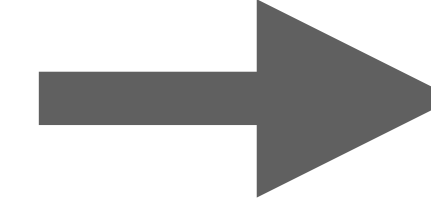
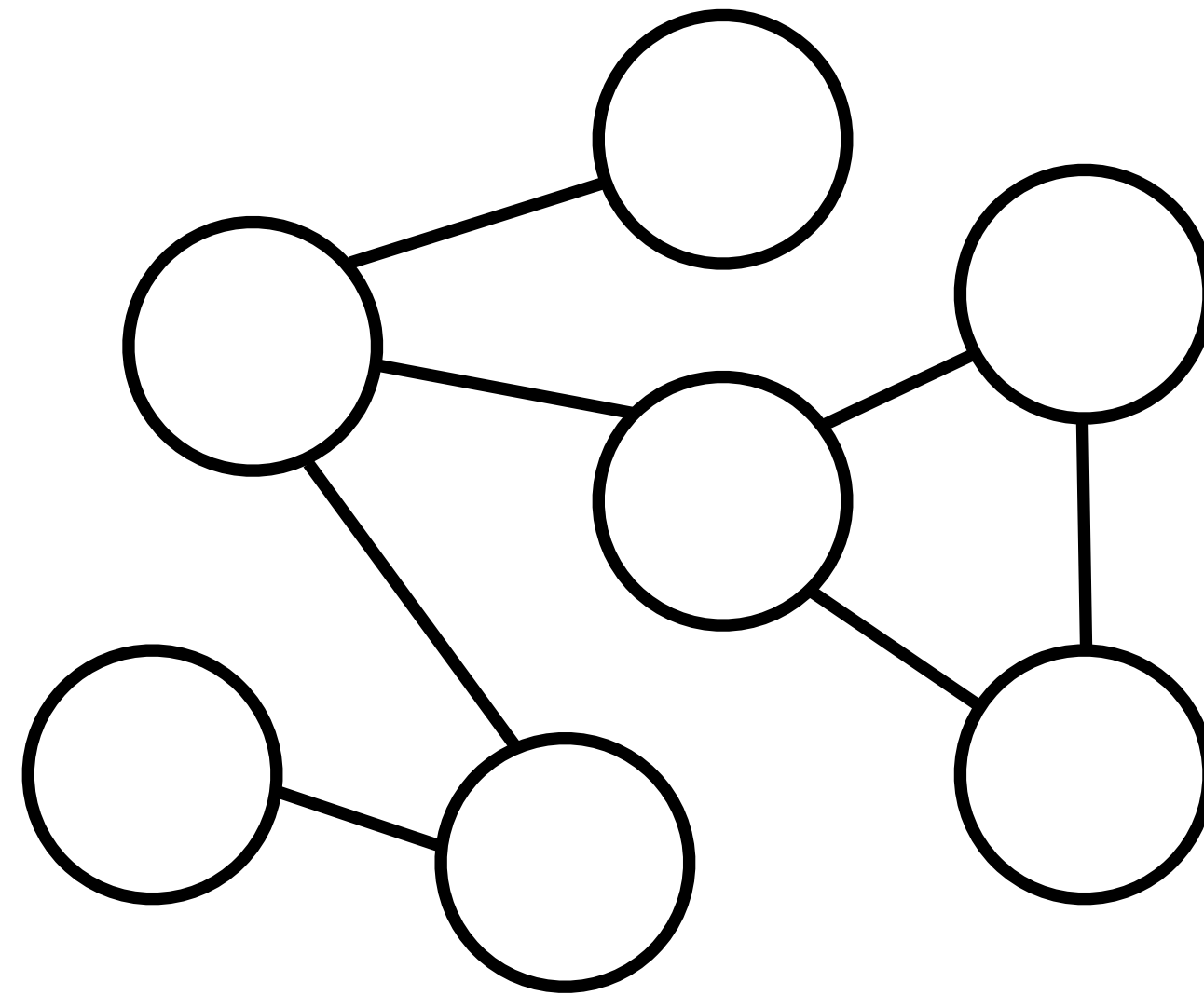
GNNs



aprenden *graph embeddings*

Graph Neural Networks

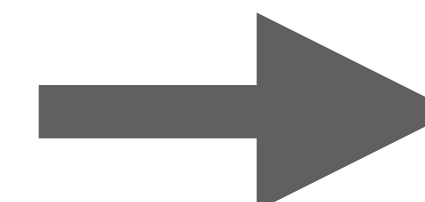
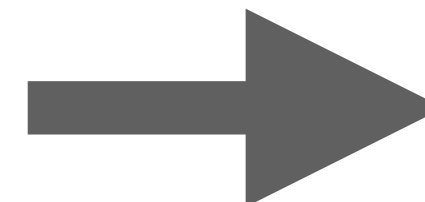
GNNs



0.22452
0.84369
0.11241
0.12459
0.72629
0.78121
0.56291

aprenden *graph embeddings*

0.22452
0.84369
0.11241
0.12459
0.72629
0.78121
0.56291



**Los usan para
tareas de ML**

Por qué graph embeddings?

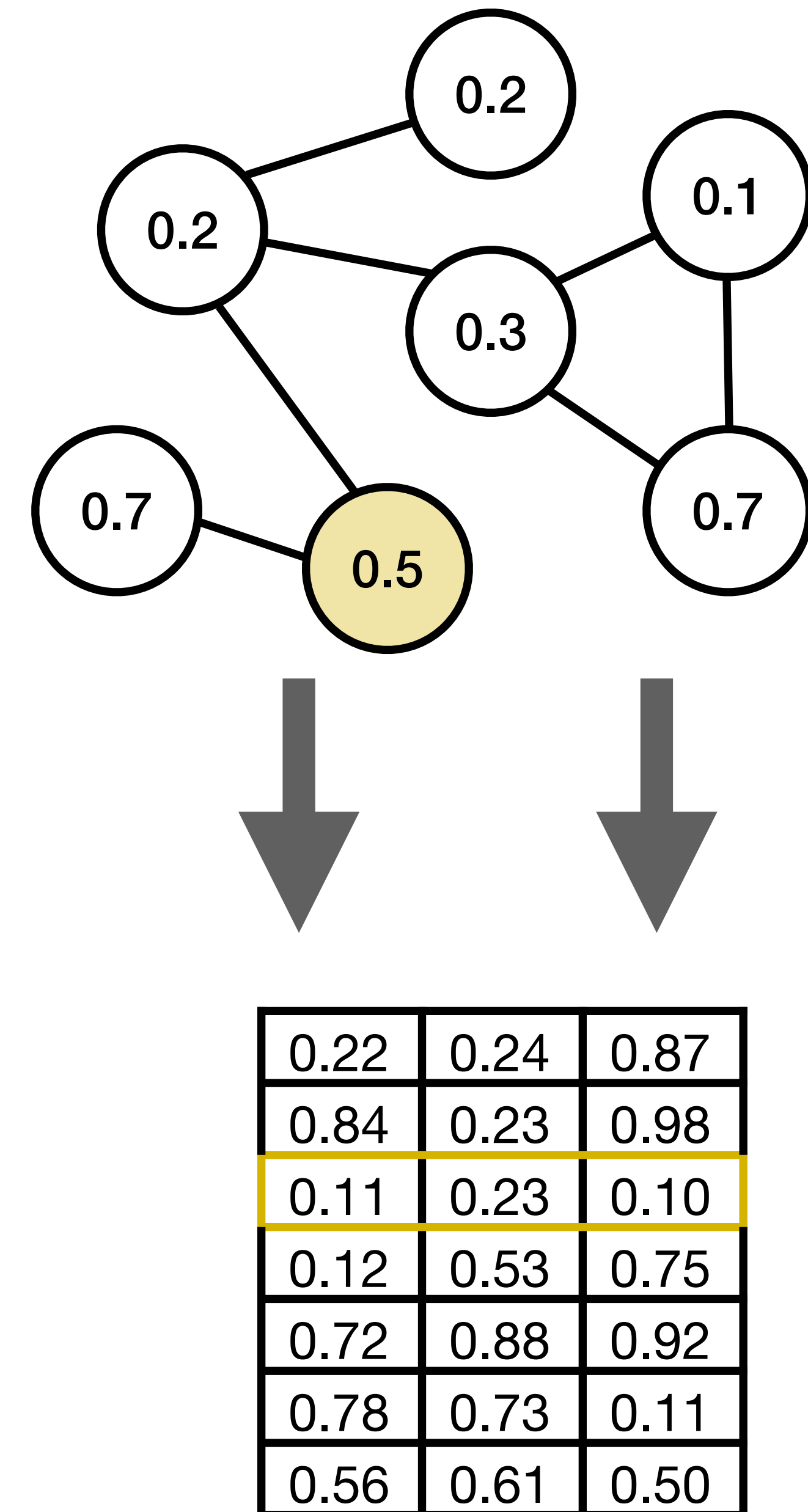
Enfoque estándar para problemas de aprendizaje en grafos:

- Clasificación de vértices
- Predicción de Links
- Regresión de valores en nodos
- Clasificación de (Sub)grafos

Construye los embeddings, luego aplica técnicas conocidas

En una GNN, los embeddings combinan datos + topología

- Los nodos tienen features
- Pero se conectan a otros nodos
- Embeddings “codifican” topología



GNNs: agrega topología a tareas de ML

- BD de un banco
- Necesitan predecir probabilidad de pago de crédito

users

id	full_name	enabled	last_login
1	John Smith	f	2017-10-25 10:26:10.015152
2	Alice Walker	t	2017-10-25 10:26:50.295461
3	Harry Potter	t	2017-10-25 10:26:50.295461
5	Jane Smith	t	2017-10-25 10:36:43.324015

books

id	title	author	published_date	isbn
1	My First SQL book	Mary Parker	2012-02-22 12:08:17.320053-03	981483029127
2	My Second SQL book	John Mayer	1972-07-03 09:22:45.050088-07	857300923713
3	My Third SQL book	Cary Flint	2015-10-18 14:05:44.547516-07	523120967812

reviews

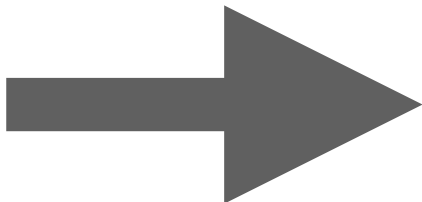
id	book_id	reviewer_name	content	rating	published_date
1	1	'John Smith'	'My first review'	4	2017-12-10 05:50:11.127281-02
2	2	'John Smith'	'My second review'	5	2017-10-13 15:05:12.673382-05
3	2	'Alice Walker'	'Another review'	1	2017-10-22 23:47:10.407569-07

checkouts

id	user_id	book_id	checkout_date	return_date
1	1	1	2017-10-15 14:43:18.095143-07	
2	1	2	2017-10-05 16:22:44.593188-07	2017-10-13 13:05:12.673382-05
3	2	2	2017-10-15 11:11:24.994973-07	2017-10-22 17:47:10.407569-07
4	5	3	2017-10-15 09:27:07.215217-07	

addresses

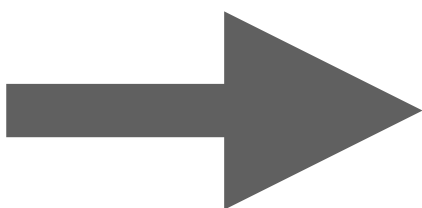
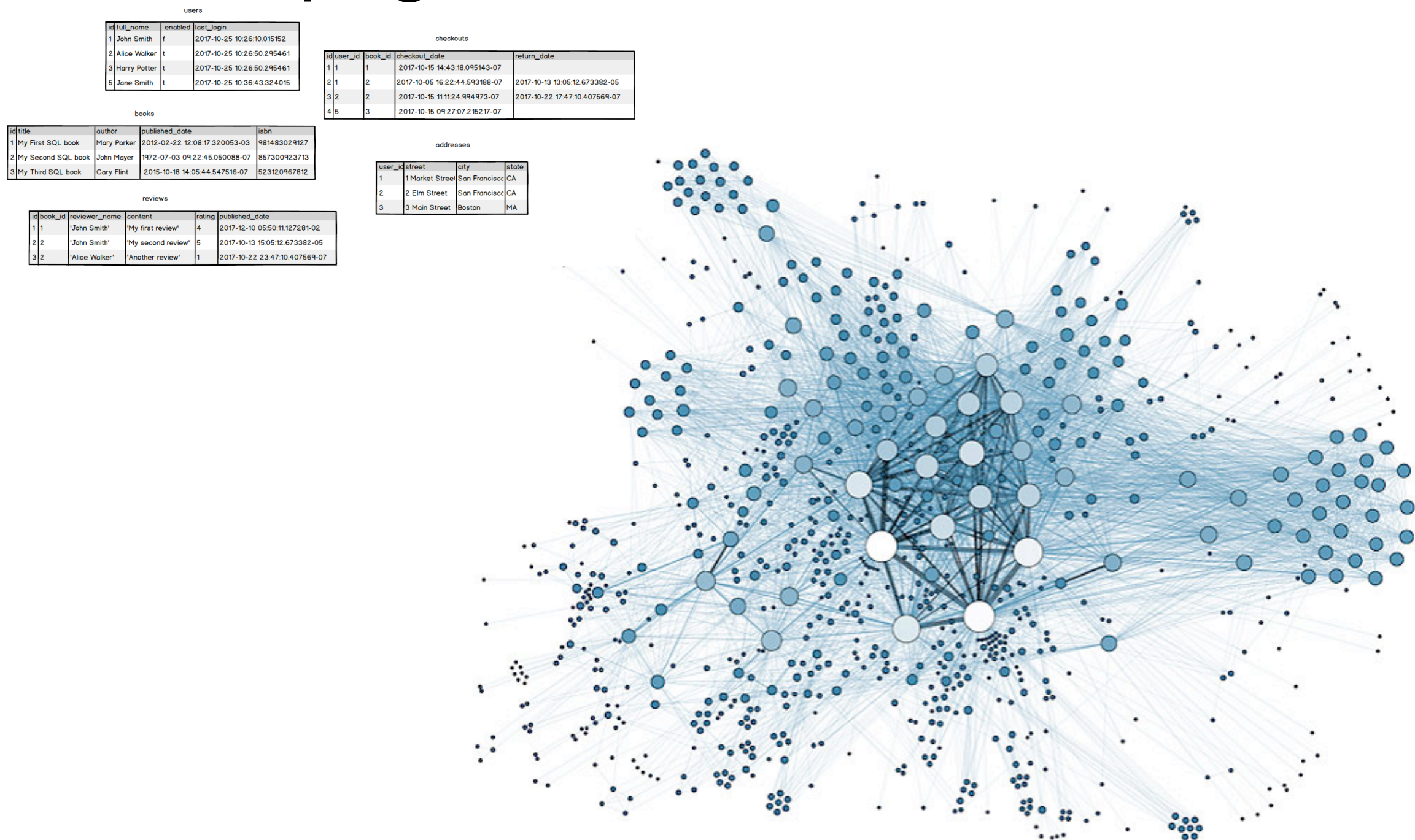
user_id	street	city	state
1	1 Market Street	San Francisco	CA
2	2 Elm Street	San Francisco	CA
3	3 Main Street	Boston	MA



Predicción

GNNs: agrega topología a tareas de ML

- BD de un banco
- Necesitan predecir probabilidad de pago de crédito
- Incorporar datos redes sociales? conexiones?



Predicción

No es llegar y meterle conexiones a un modelo

Idea: agregar tabla indicando aristas.

Correr algún modelo de deep learning.

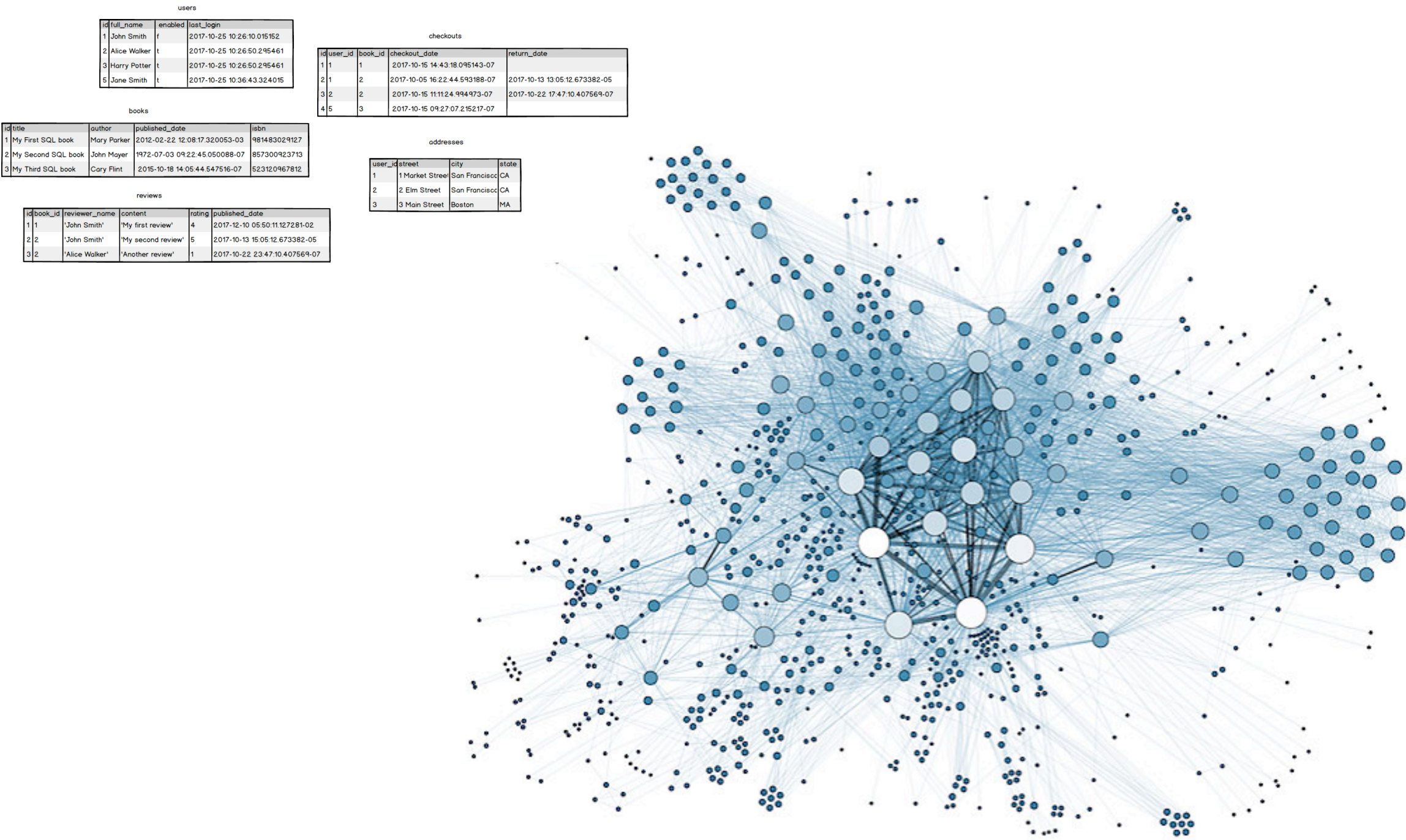


Image credit: AURAI

No es llegar y meterle conexiones a un modelo

Idea: agregar tabla indicando aristas.
Correr algún modelo de deep learning.

users			
id	full_name	enabled	last_login
1	John Smith	t	2017-10-25 10:26:10.015152
2	Alice Walker		2017-10-25 10:26:50.2195461
3	Harry Potter		2017-10-25 10:26:50.2195461
5	Jane Smith	t	2017-10-25 10:36:43.324015

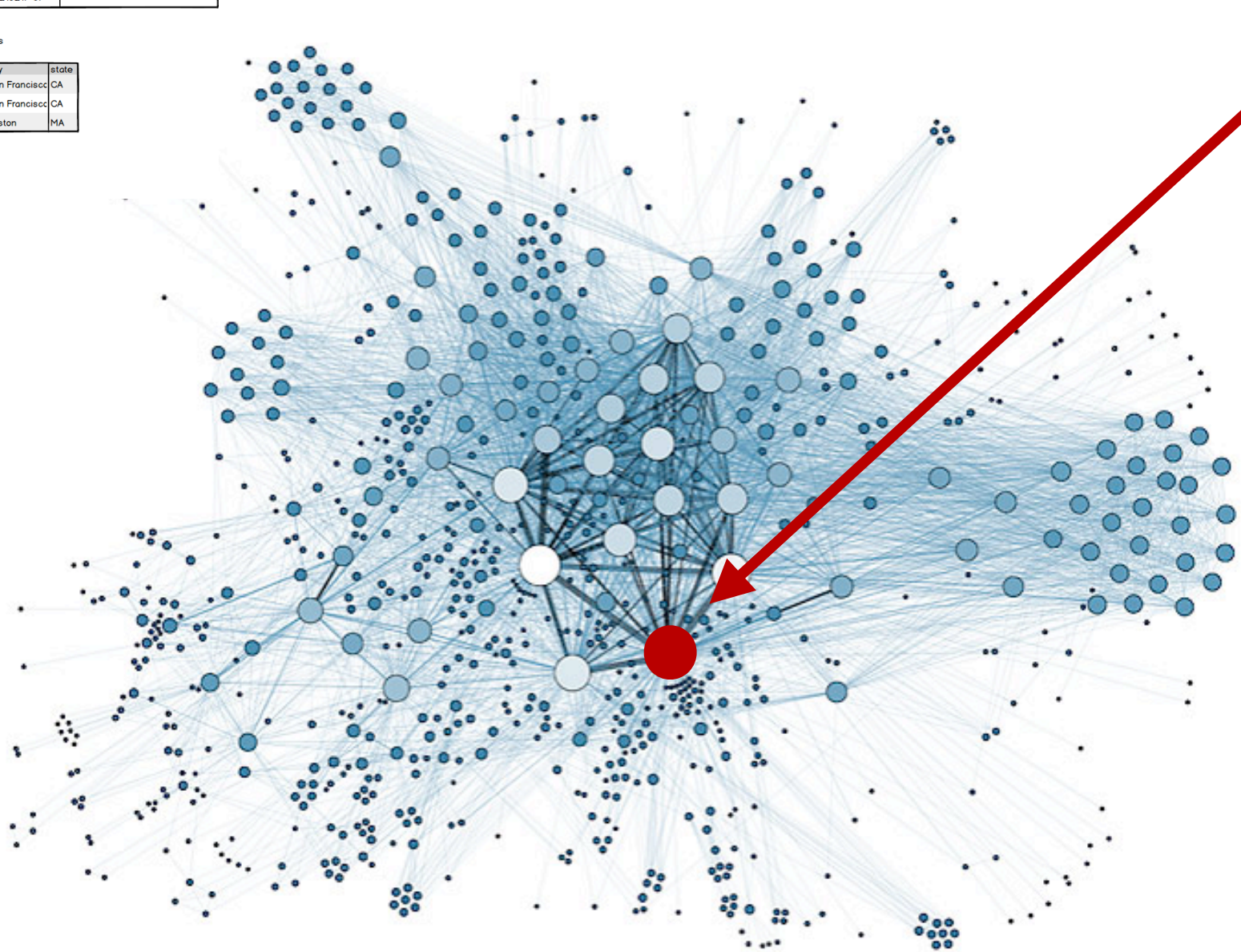
books			
id	title	author	published_date
1	My First SQL book	Mary Parker	2012-02-22 12:08:17.320053-03
2	My Second SQL book	John Mayer	1972-07-03 09:22:45.050089-07
3	My Third SQL book	Cary Flint	2015-10-18 14:05:44.547516-07

checkouts			
user_id	book_id	checkout_date	return_date
1	1	2017-10-15 14:43:18.0195143-07	
2	1	2017-10-05 16:22:44.5193188-07	2017-10-13 13:05:12.673382-05
3	2	2017-10-15 11:11:24.9141973-07	
4	5	2017-10-15 09:27:07.215217-07	

addresses			
user_id	street	city	state
1	1 Market Street	San Francisco	CA
2	2 Elm Street	San Francisco	CA
3	3 Main Street	Boston	MA

reviews				
id	book_id	reviewer_name	content	rating
1	1	John Smith	My first review	4
2	2	John Smith	My second review	5
3	2	Alice Walker	Another review	1

Este nodo puede tener un montón de amigos



No es llegar y meterle conexiones a un modelo

Idea: agregar tabla indicando aristas.

Correr algún modelo de deep learning.

users			
id	full_name	enabled	last_login
1	John Smith	t	2017-10-25 10:26:10.015152
2	Alice Walker		2017-10-25 10:26:50.2195461
3	Harry Potter		2017-10-25 10:26:50.2195461
5	Jane Smith	t	2017-10-25 10:36:43.324015

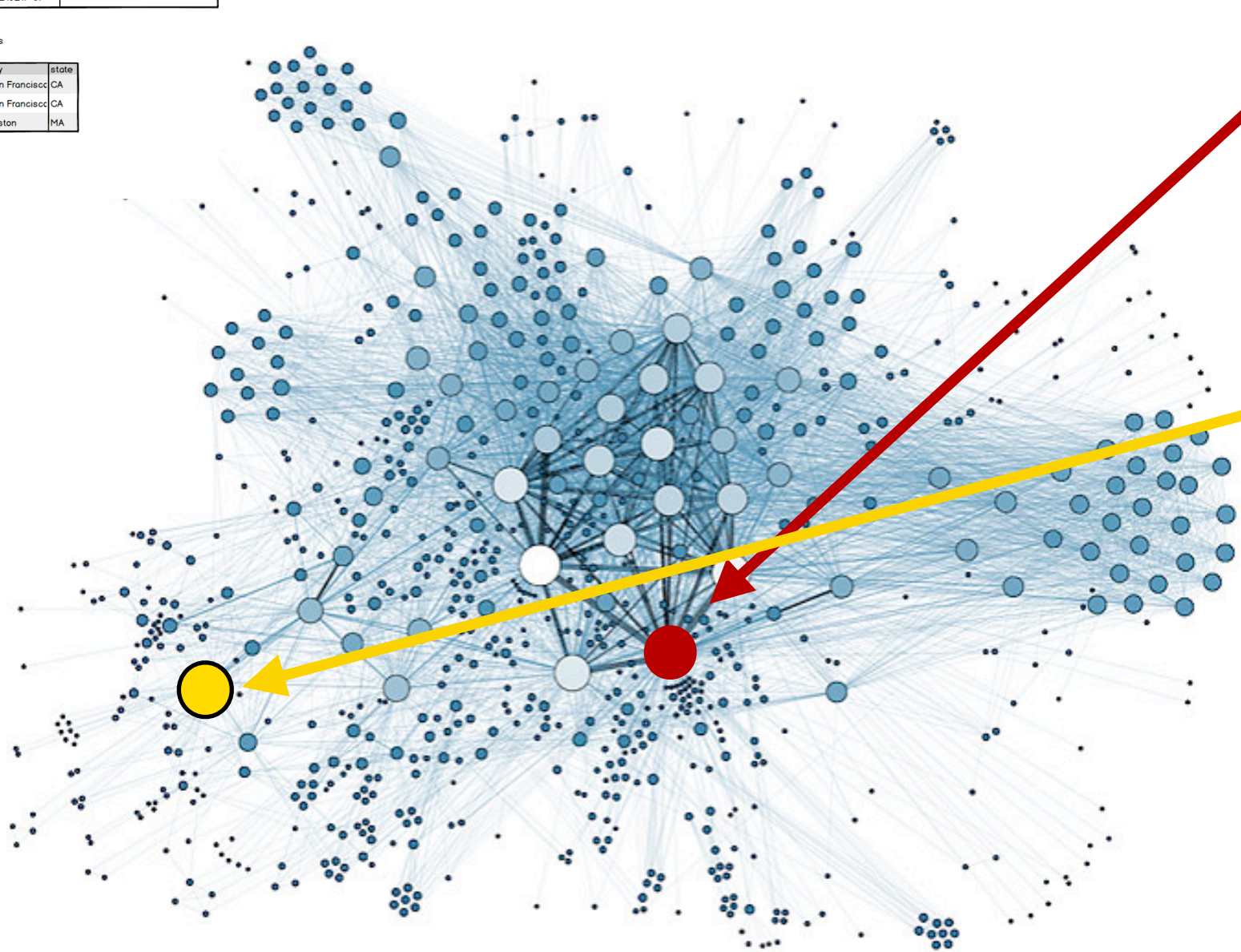
books			
id	title	author	published_date
1	My First SQL book	Mary Parker	2012-02-22 12:08:17.320053-03
2	My Second SQL book	John Mayer	1972-07-03 09:22:45.050088-07
3	My Third SQL book	Cary Flint	2015-10-18 14:05:44.547516-07

checkouts			
user_id	book_id	checkout_date	return_date
1	1	2017-10-15 14:43:18.0195143-07	
2	1	2017-10-05 16:22:44.5193188-07	2017-10-13 13:05:12.673382-05
3	2	2017-10-15 11:1124.9141973-07	
4	5	2017-10-15 09:27:07.215217-07	

addresses			
user_id	street	city	state
1	1 Market Street	San Francisco	CA
2	2 Elm Street	San Francisco	CA
3	3 Main Street	Boston	MA

reviews				
id	book_id	reviewer_name	content	rating
1	1	'John Smith'	'My first review'	4
2	2	'John Smith'	'My second review'	5
3	2	'Alice Walker'	'Another review'	1

Este nodo puede tener un montón de amigos



Este es importante para la predicción del rojo, pero no están conectados

No es llegar y meterle conexiones a un modelo

Idea: agregar tabla indicando aristas.

Correr algún modelo de deep learning.

users			
id	full_name	enabled	last_login
1	John Smith	t	2017-10-25 10:26:10.015152
2	Alice Walker	t	2017-10-25 10:26:50.2195461
3	Harry Potter	t	2017-10-25 10:26:50.2195461
5	Jane Smith	t	2017-10-25 10:36:43.324015

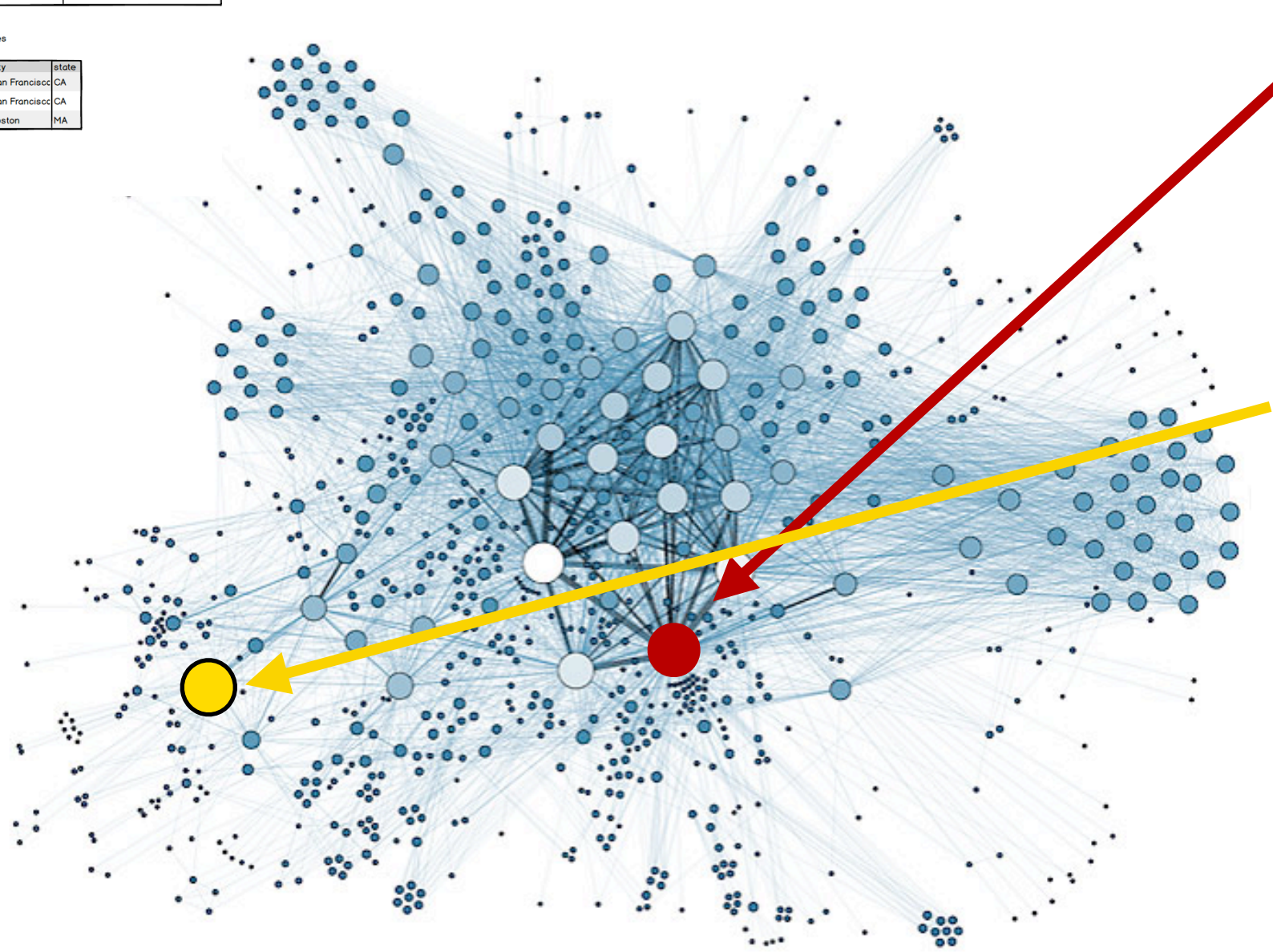
checkouts			
user_id	book_id	checkout_date	return_date
1	1	2017-10-15 14:43:18.0195143-07	
2	1	2017-10-05 16:22:44.5193188-07	2017-10-13 13:05:12.673382-05
3	2	2017-10-15 11:11:24.9141973-07	
4	5	2017-10-15 09:27:07.215217-07	

books				
id	title	author	published_date	isbn
1	My First SQL book	Mary Parker	2012-02-22 12:08:17.320053-03	981483021127
2	My Second SQL book	John Mayer	1972-07-03 09:22:45.050088-07	857300923713
3	My Third SQL book	Cary Flint	2015-10-18 14:05:44.547516-07	523120167812

addresses			
user_id	street	city	state
1	1 Market Street	San Francisco	CA
2	2 Elm Street	San Francisco	CA
3	3 Main Street	Boston	MA

reviews				
id	book_id	reviewer_name	content	rating
1	1	'John Smith'	'My first review'	4
2	2	'John Smith'	'My second review'	5
3	2	'Alice Walker'	'Another review'	1

Este nodo puede tener un montón de amigos



Este es importante para la predicción del rojo, pero no están conectados

N nodos -> matriz de tamaño N^2!

Can't just add the data to chatGPT either



GPT-4 tiene ventana de 32000 tokens

DBLP tiene 16,689,230 aristas

Can't just add the data to chatGPT either



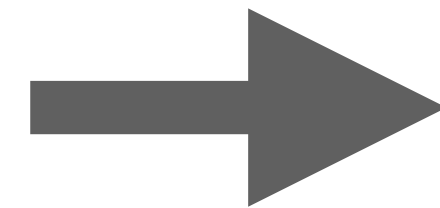
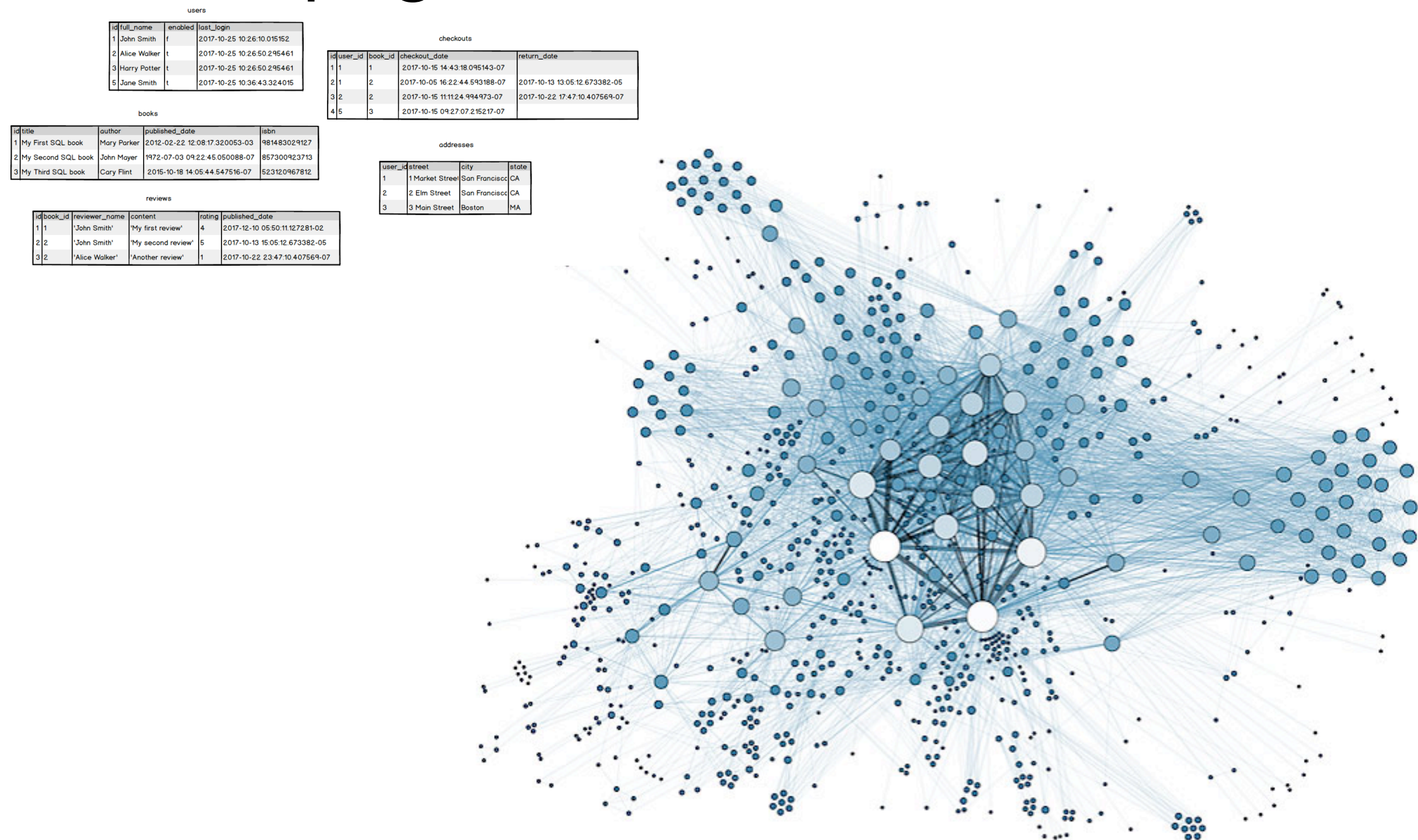
GPT-4 tiene ventana de 32000 tokens

DBLP tiene 16,689,230 aristas

También: nos gustaría no depender del orden en que nos pasan el grafo.

GNNs: agrega topología a tareas de ML

- BD de un banco
- Necesitan predecir probabilidad de pago de crédito
- Incorporar datos redes sociales? conexiones?



Predicción

¿Cómo hacemos esto? GNNs!

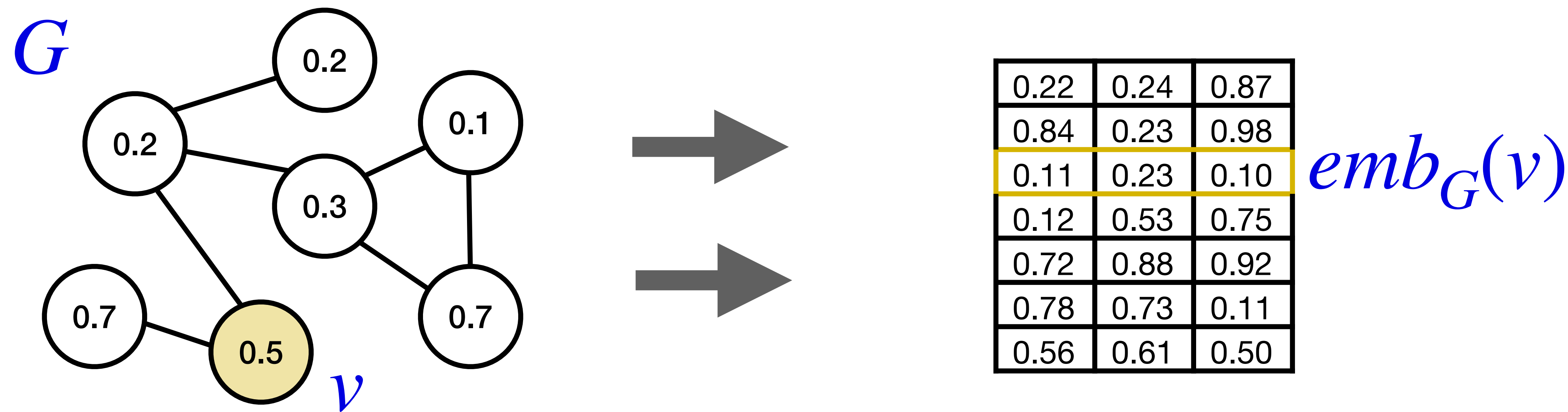
Mirando como se ve una arquitectura básica de GNN

Mirando como se ve una arquitectura básica de GNN

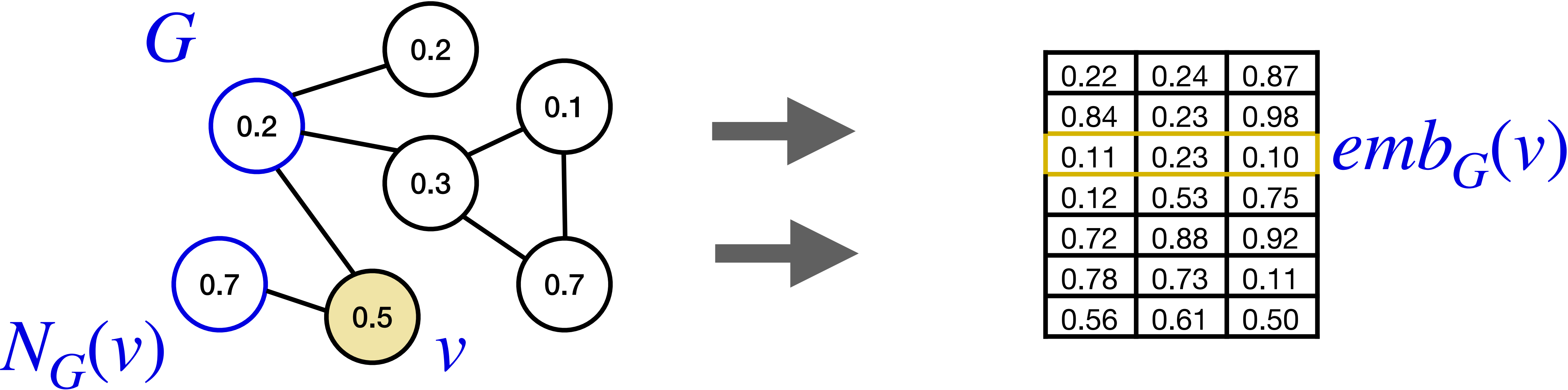
El bloque fundamental de una GNNs se conoce como una capa de
Message Passing

Las arquitecturas que solo usan esas capas son
Message Passing Neural Networks (MPNNs)

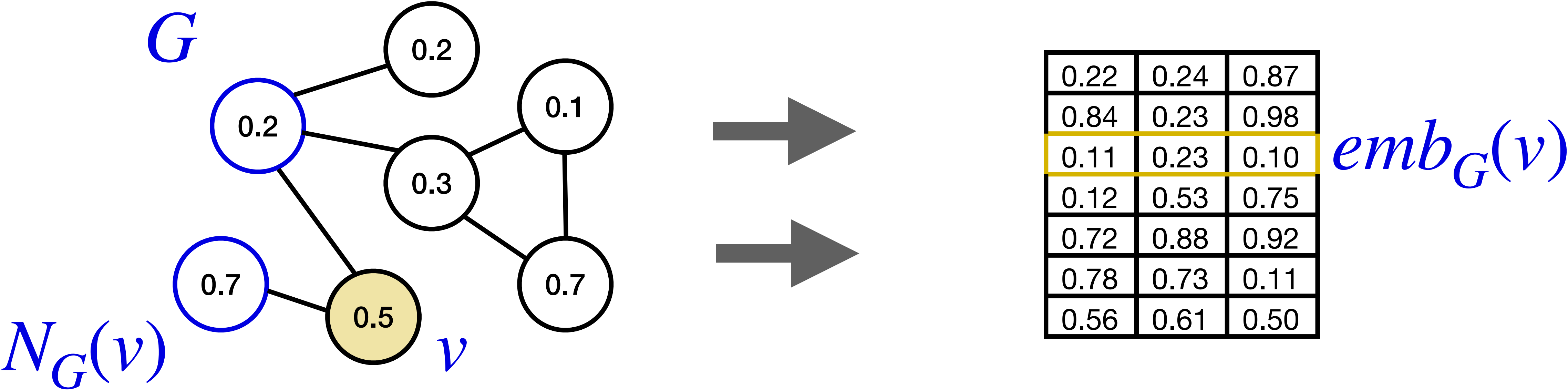
Capa MPNN (Message Passing Neural Network)



Capa MPNN (Message Passing Neural Network)



Capa MPNN (Message Passing Neural Network)



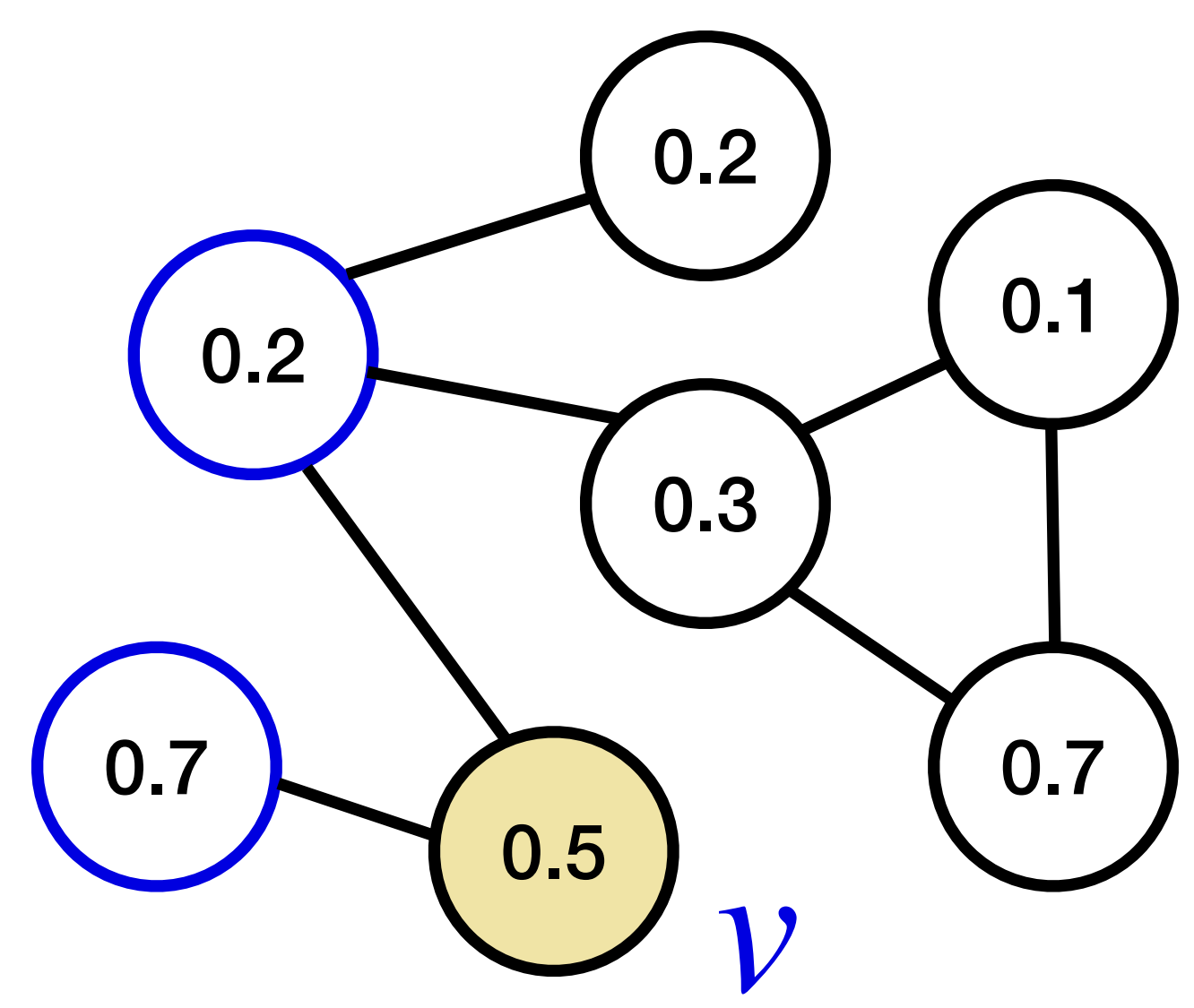
Nuevos embeddings $emb_G(v)$
De embeddings antiguos $emb'_G(v)$

Capa MPNN (Message Passing Neural Network)

Nuevos embeddings $emb_G(v)$
De embeddings antiguos $emb'_G(v)$

$$\text{Aggregate}\left(\left\{\left\{ emb'_G(w) \mid w \in N_G(v) \right\}\right\}\right)$$

$$emb_G(v) \text{ usa } \text{Aggregate}(0.7, 0.2)$$



Capa MPNN (Message Passing Neural Network)

Nuevos embeddings $emb_G(v)$
De embeddings antiguos $emb'_G(v)$

$$\text{Aggregate}\left(\left\{\left\{ emb'_G(w) \mid w \in N_G(v) \right\}\right\}\right)$$

Aggregate es una function que agrega embeddings de vecinos.
¡Debe ser **invariante a permutaciones en la matriz de adyacencia**!

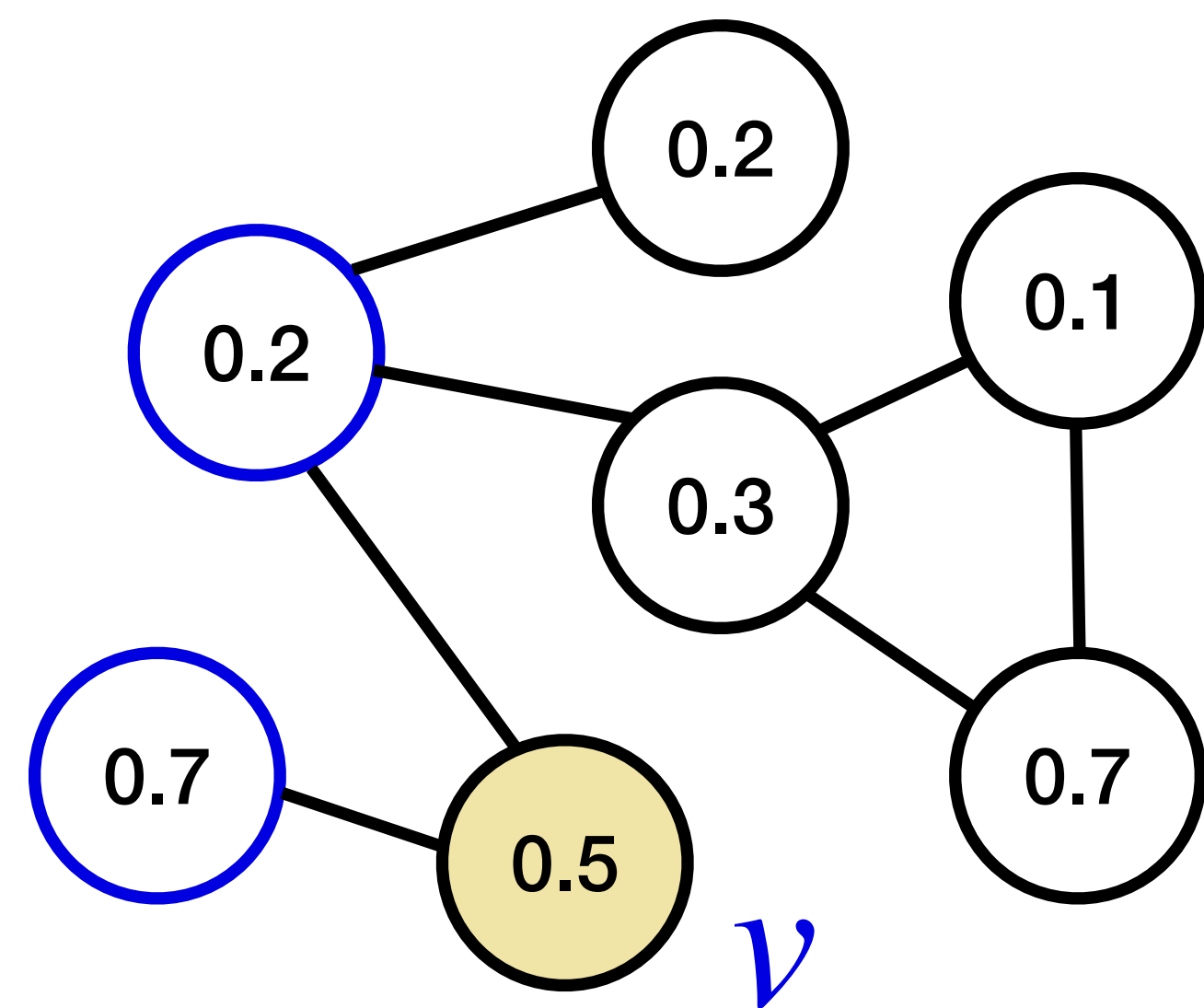
Sum, Max, Avg, etc. Puede incorporar pesos y otras cosas entrenables.

Capa MPNN (Message Passing Neural Network)

Nuevos embeddings $emb_G(v)$
De embeddings antiguos $emb'_G(v)$

$$\text{Aggregate}\left(\left\{\left\{ emb'_G(w) \mid w \in N_G(v) \right\}\right\}\right)$$

$$emb_G(v) \text{ usa } \text{Sum}\left(MLP(0.7), MLP(0.2)\right)$$



Capa MPNN (Message Passing Neural Network)

Nuevos embeddings $emb_G(v)$
De embeddings antiguos $emb'_G(v)$

$$\text{Update}\left(emb'_G(v), \text{Aggregate}\left(\{\{emb'_G(w) \mid w \in N_G(v)\}\}\right)\right)$$

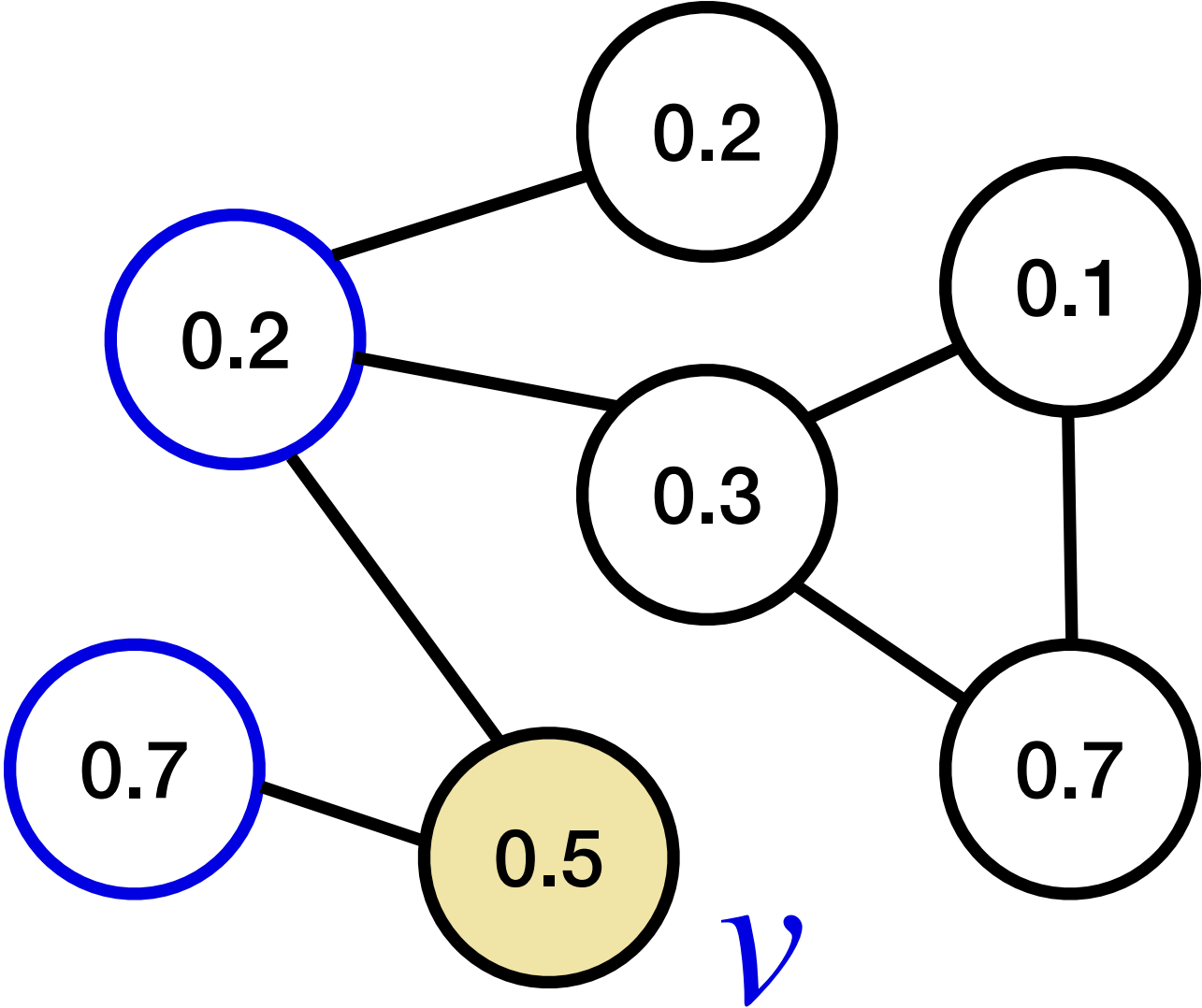
Update es una función que combina mi embedding con el agregado.

Concat, **Sum**, etc. Puede incorporar pesos entrenables.

Capa MPNN (Message Passing Neural Network)

Nuevos embeddings $emb_G(v)$
De embeddings antiguos $emb'_G(v)$

$$\text{Update}\left(emb'_G(v), \text{Aggregate}\left(\{\{emb'_G(w) \mid w \in N_G(v)\}\}\right)\right)$$



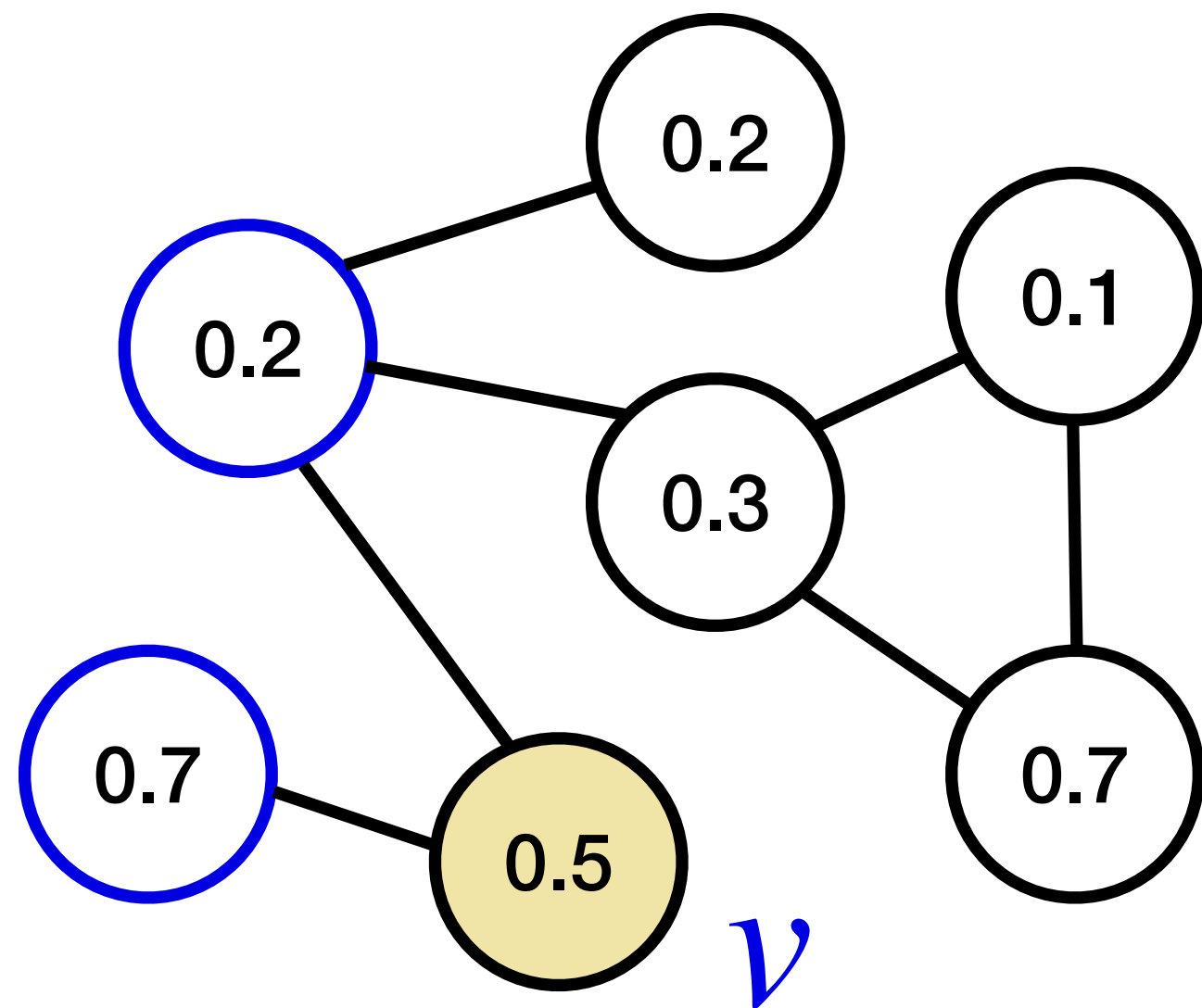
$emb_G(v)$ usa :

$$\text{Update}\left(0.5, \text{Aggregate}(0.7, 0.2)\right)$$

Capa MPNN (Message Passing Neural Network)

Nuevos embeddings $emb_G(v)$
De embeddings antiguos $emb'_G(v)$

$$\text{Update}\left(\textcolor{brown}{emb}'_G(v), \text{Aggregate}\left(\{\{\textcolor{brown}{emb}'_G(w) \mid w \in N_G(v)\}\}\right)\right)$$



$emb_G(v)$ usa :

Update $\left(MLP(0.5), \text{Sum}(MLP(0.7), MLP(0.2)) \right)$

Capa MPNN (Message Passing Neural Network)

Nuevos embeddings $emb_G(v)$
De embeddings antiguos $emb'_G(v)$

$$emb_G(v) := \sigma \left(\text{Update}(emb'_G(v), \text{Aggregate}(\{ \{ emb'_G(w) \mid w \in N_G(v) \} \})) \right)$$

El resultado de Aggregate - update se pasa por una activación no lineal

Capa MPNN (Message Passing Neural Network)

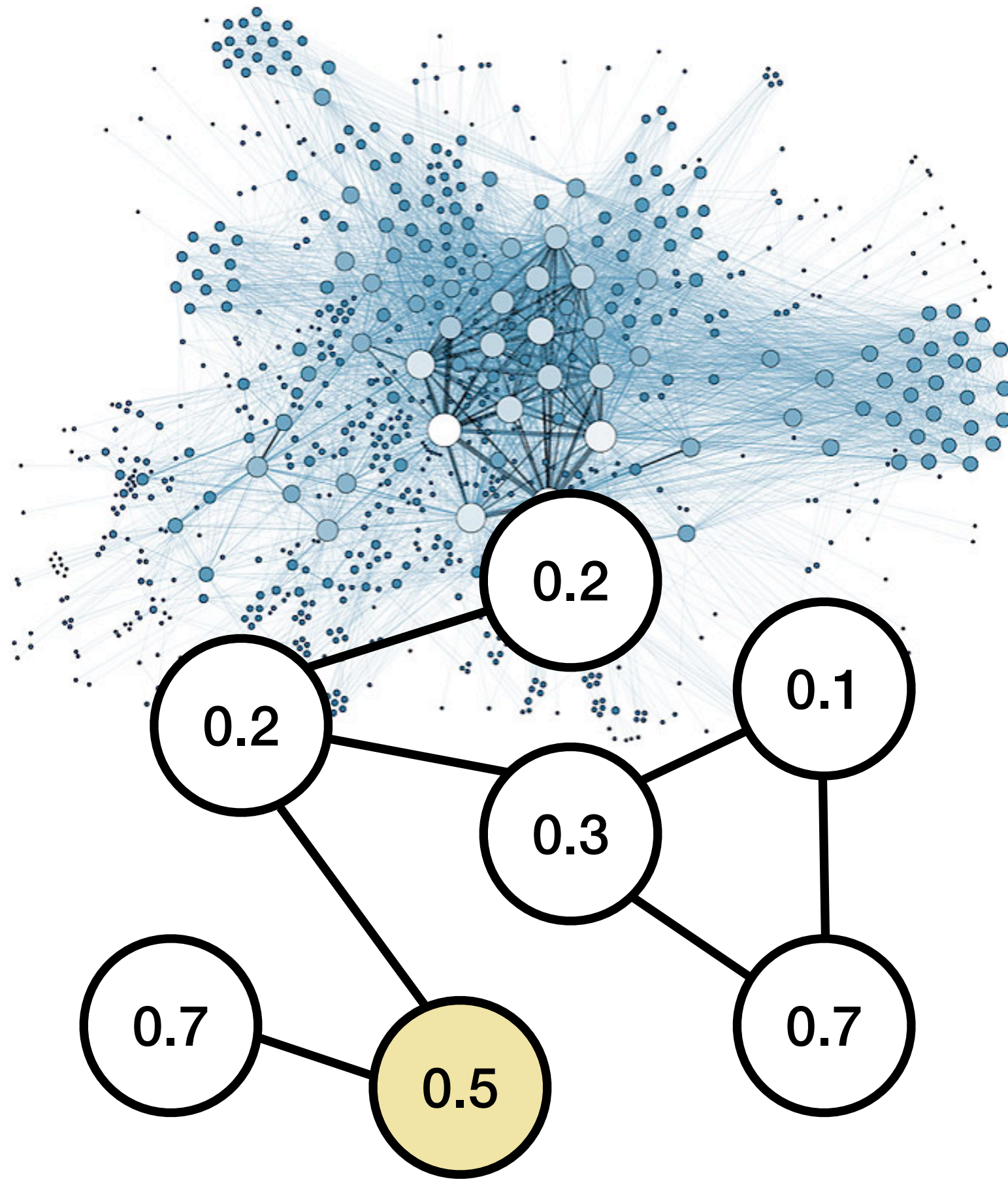
Nuevos embeddings $emb_G(v)$
De embeddings antiguos $emb'_G(v)$

1. **Agregar** la información de mis vecinos
2. Hacer **Update** del agregado con lo que tenía antes
3. Pasar todo por una **función de activación no lineal**

Capa MPNN (Message Passing Neural Network)

Forward Propagation

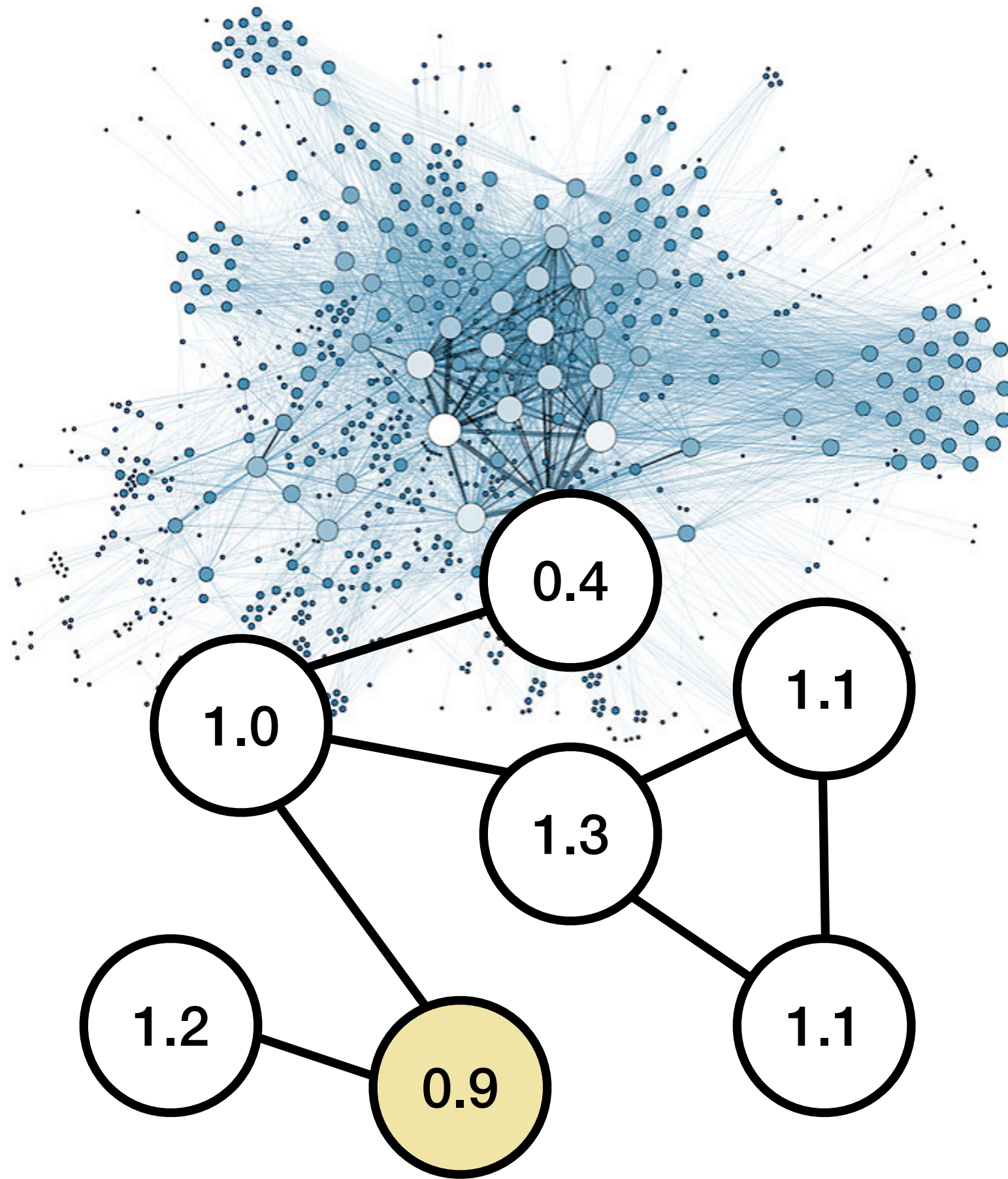
- Nodos con features
- Capa i : todos los nodos se actualizan usando los valores de la capa ($i-1$)



Capa MPNN (Message Passing Neural Network)

Forward Propagation

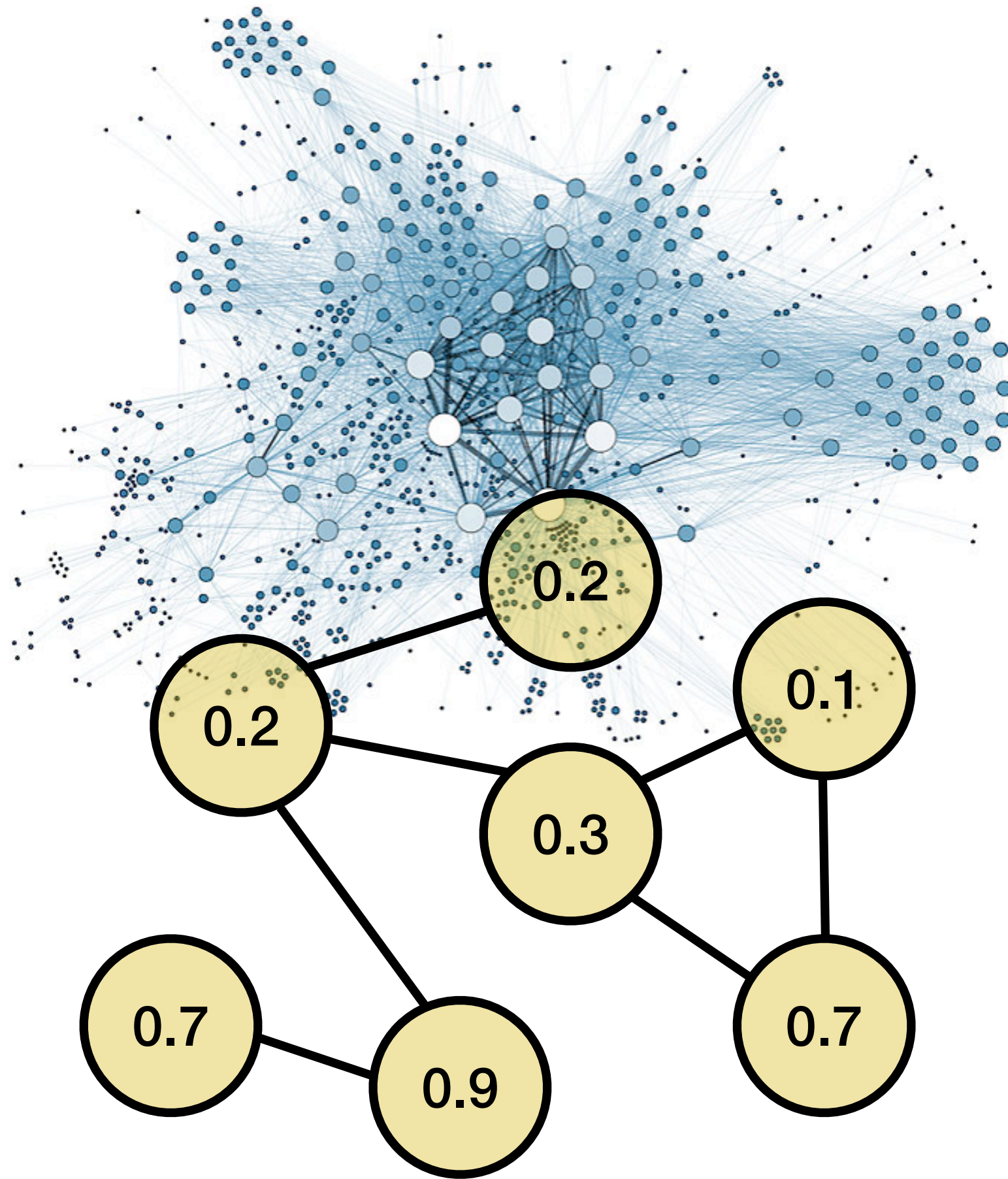
- Nodos con features
- Capa i : todos los nodos se actualizan usando los valores de la capa ($i-1$)



Capa MPNN (Message Passing Neural Network)

Forward Propagation

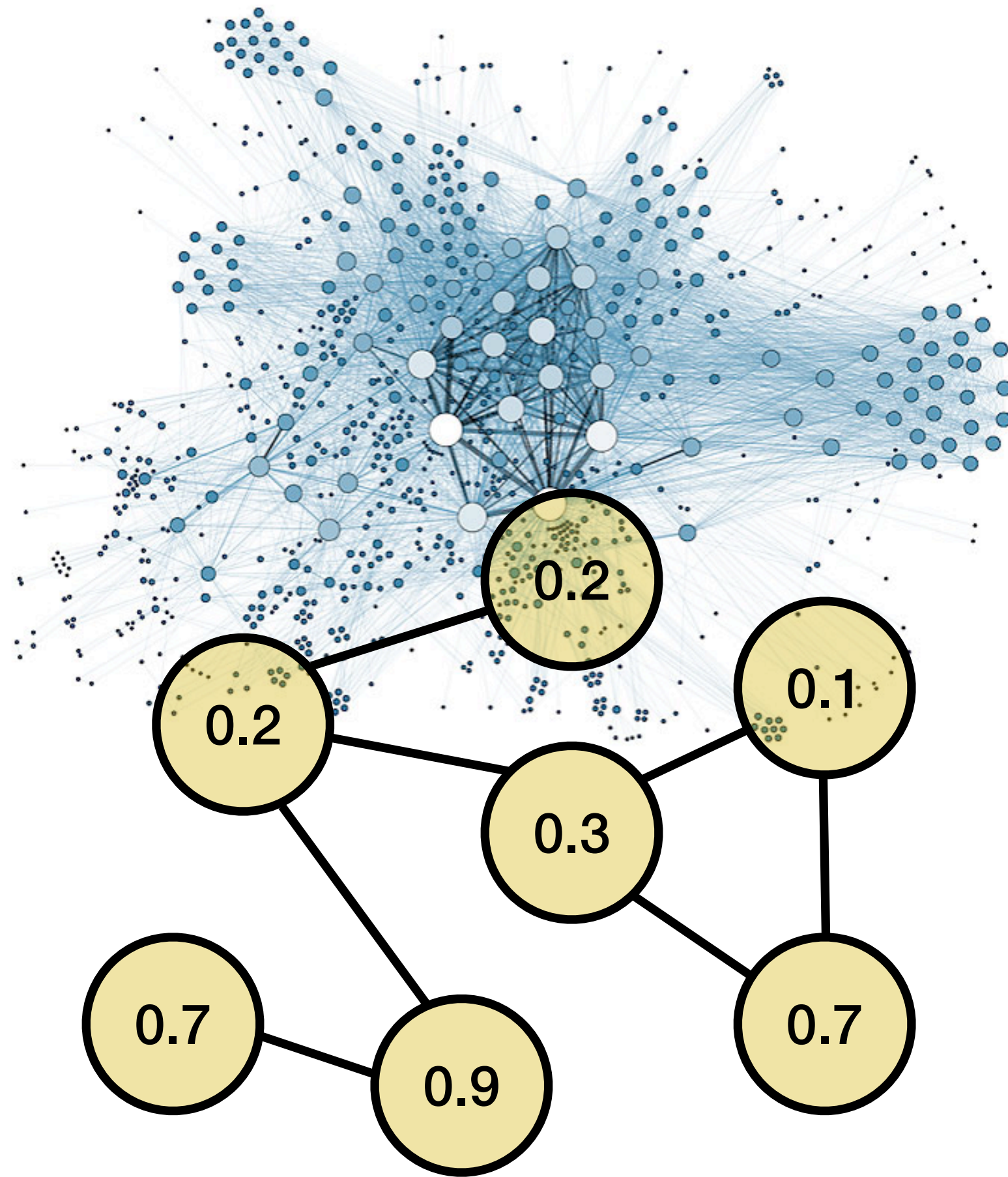
- Nodos con features
- Capa i : todos los nodos se actualizan usando los valores de la capa ($i-1$)



Entrenando capas MPNN (caso supervisado)

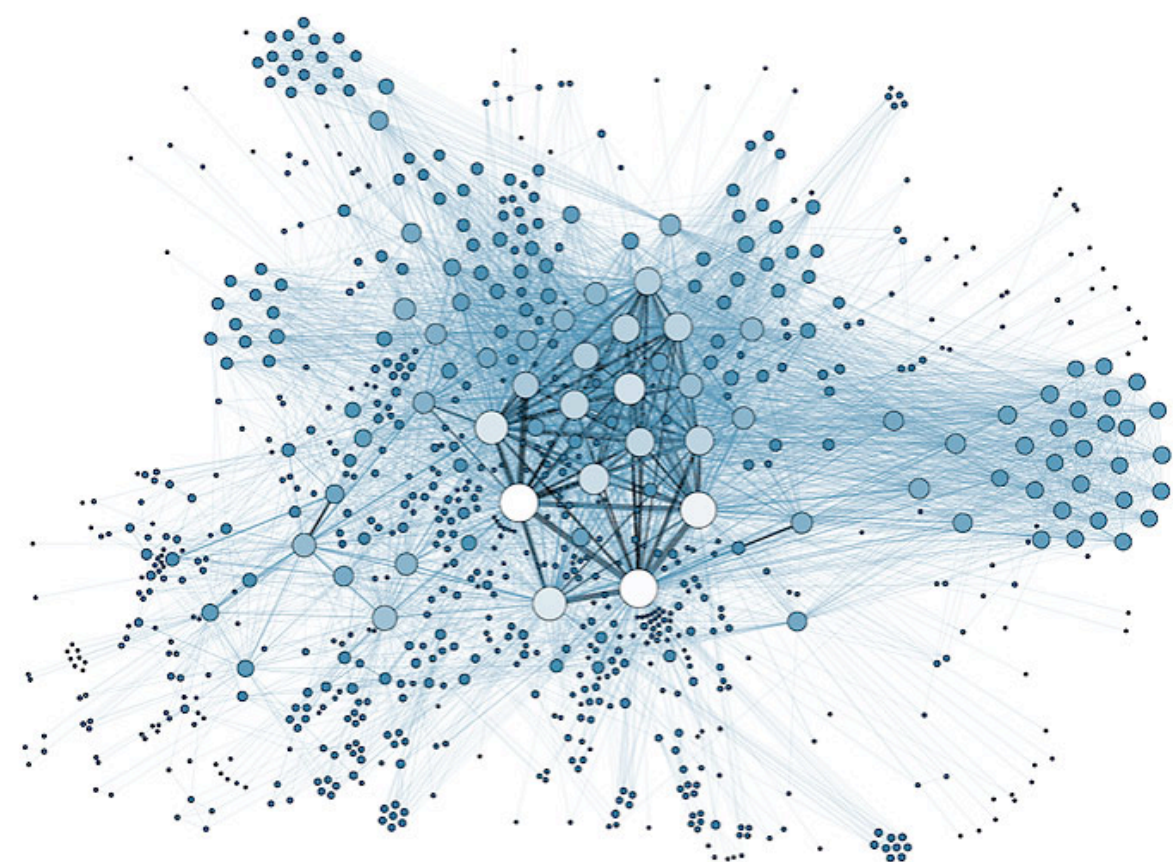
Forward Propagation

- Nodos con features
- Capa i : todos los nodos se actualizan usando los valores de la capa ($i-1$)
- Capa final de clasificación (nodo o arista)



Predicción

Entrenando capas MPNN (caso supervisado)

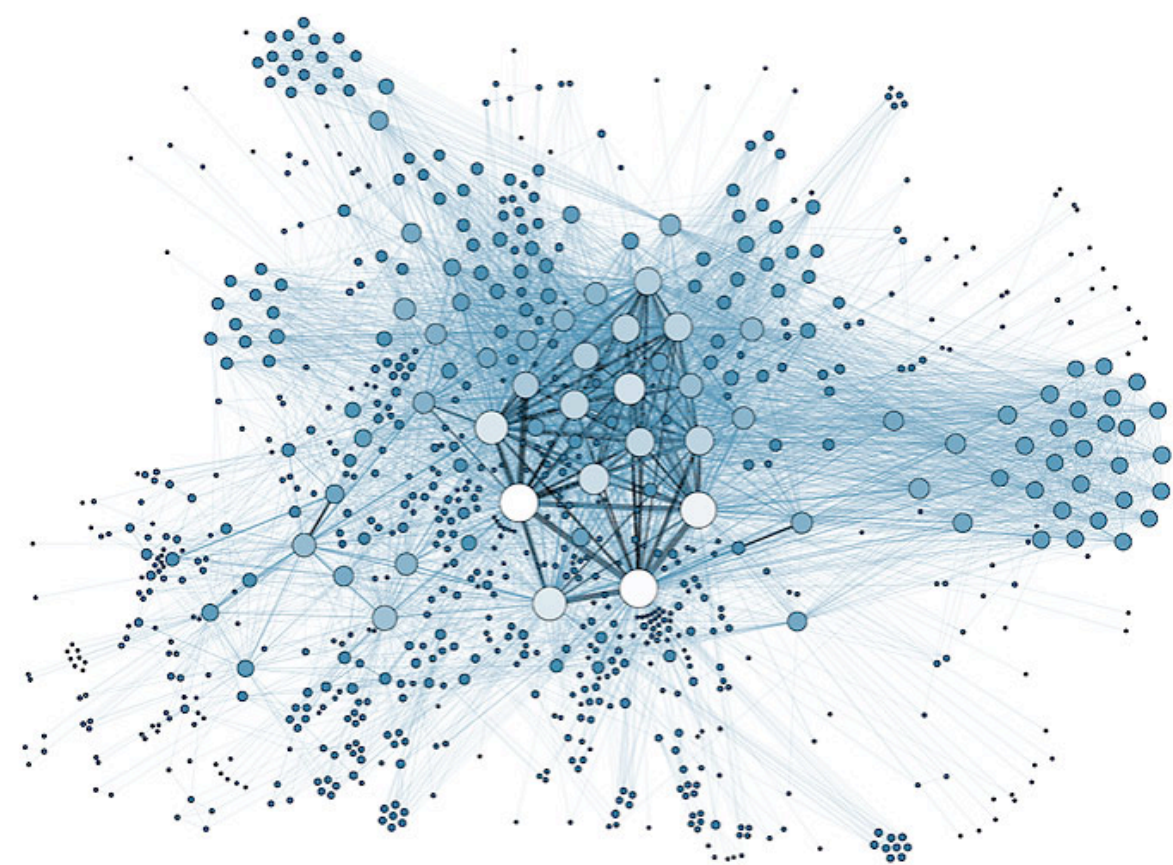


- Conocemos el valor correcto de algunos nodos.
- Hacemos Forward Propagation sobre todos los nodos.
- Solo hacemos back propagation con los nodos que conocemos el valor correcto.

Nodes
1
2
3
4
...
...
...
1000020
1000021
1000022

Training	Value
1	1
2	0
4	0
...	...
1000020	1
1000022	1

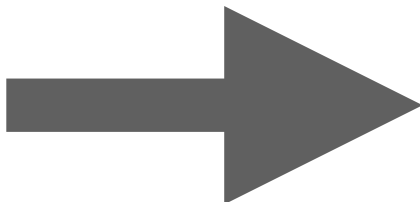
Entrenando capas MPNN (caso supervisado)



- Conocemos el valor correcto de algunos nodos.
- Hacemos Forward Propagation sobre todos los nodos.
- Solo hacemos back propagation con los nodos que conocemos el valor correcto.

Nodes
1
2
3
4
...
...
...
1000020
1000021
1000022

Training	Value
1	1
2	0
4	0
...	...
1000020	1
1000022	1



Training	Value
1	0.84369
2	0.11241
4	0.12459
...	...
1000020	0.78121
1000022	0.56291

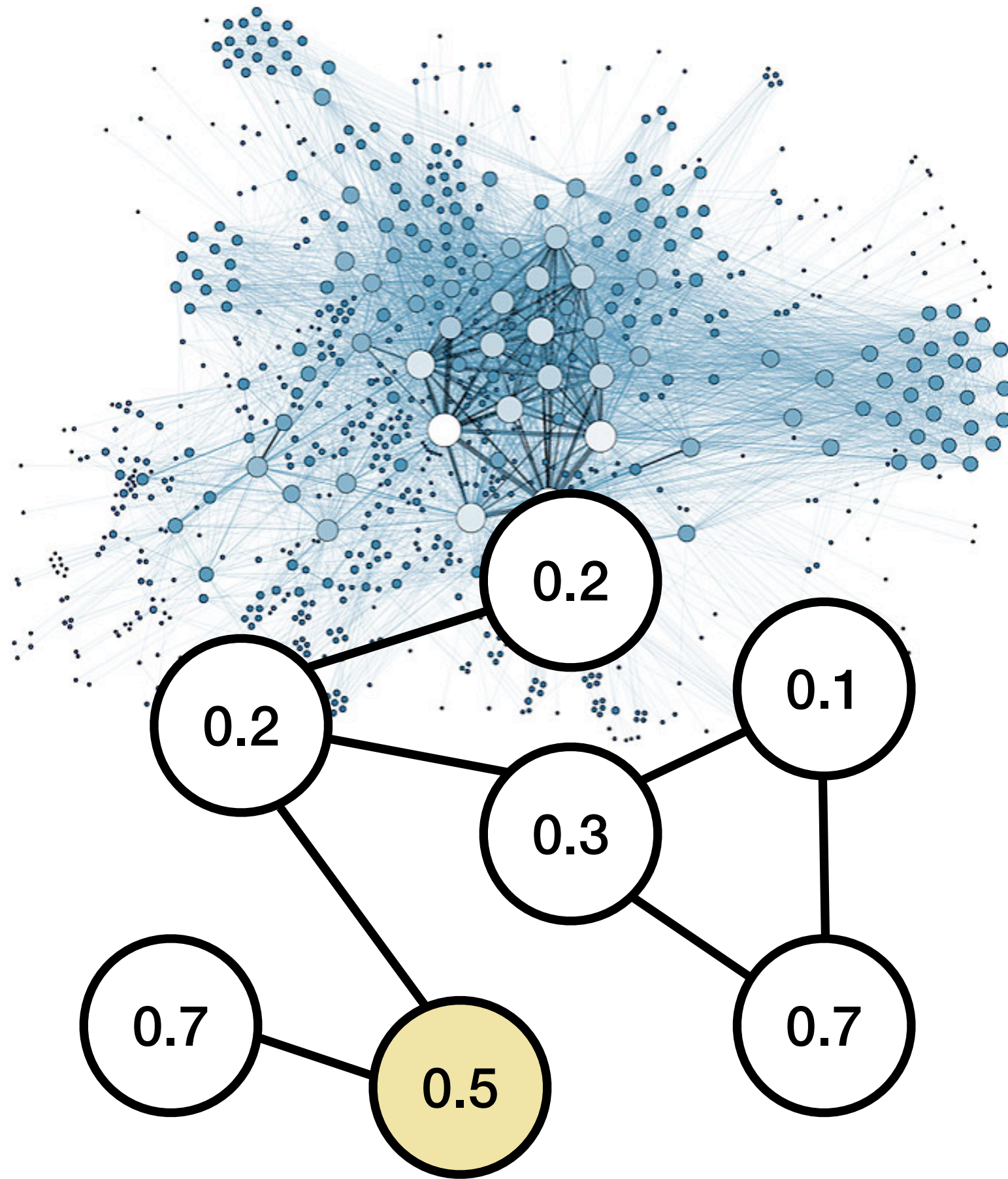
v/s

Y
1
0
0
...
1
1

Entrenando capas MPNN (caso self-supervised)

Forward Propagation

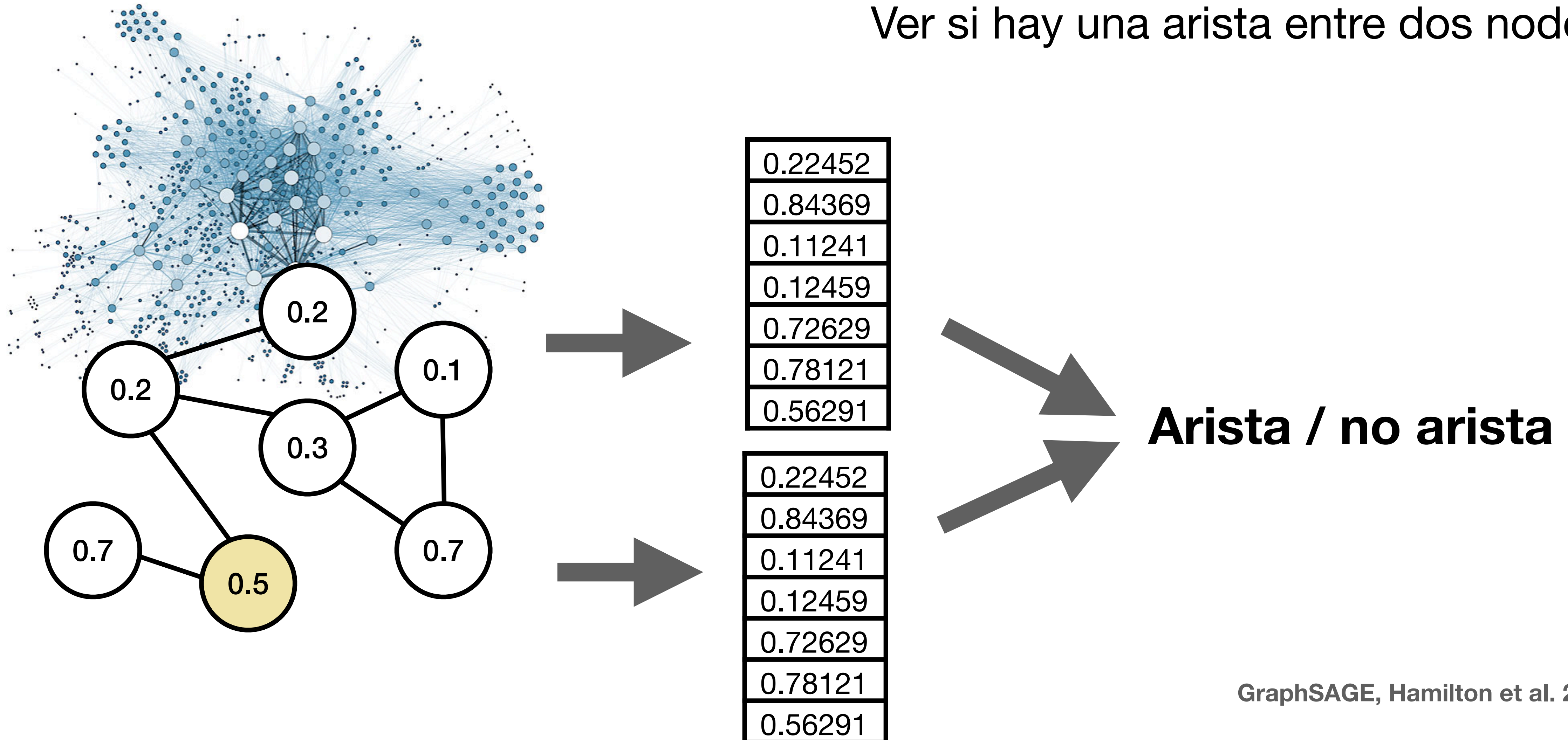
- Conocemos el valor correcto de algunos nodos.
- Hacemos Forward Propagation sobre todos los nodos.
- Capa final: **self supervised task**, p. Ej. Ver si hay una arista entre dos nodos.



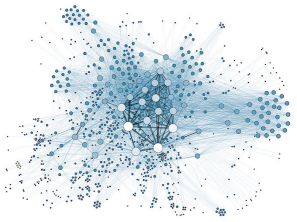
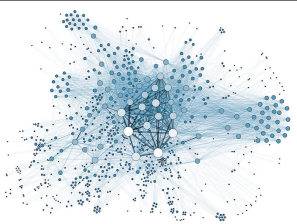
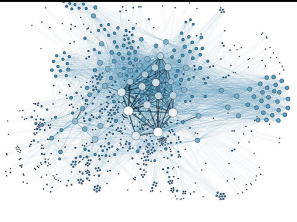
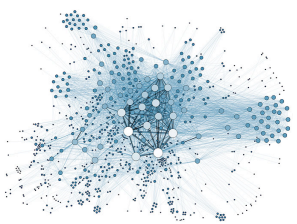
Predicción

Entrenando capas MPNN (caso self-supervised)

- Capa final: **self supervised task**, p. Ej. Ver si hay una arista entre dos nodos.

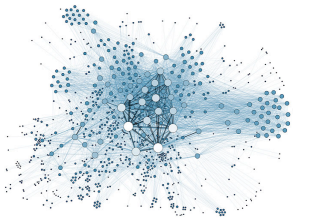
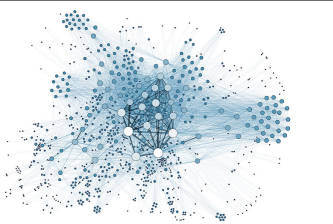
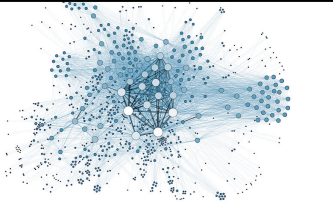



Capas MPNN para clasificar grafos

Training	Value
	1
	0
...	...
...	...
	
	

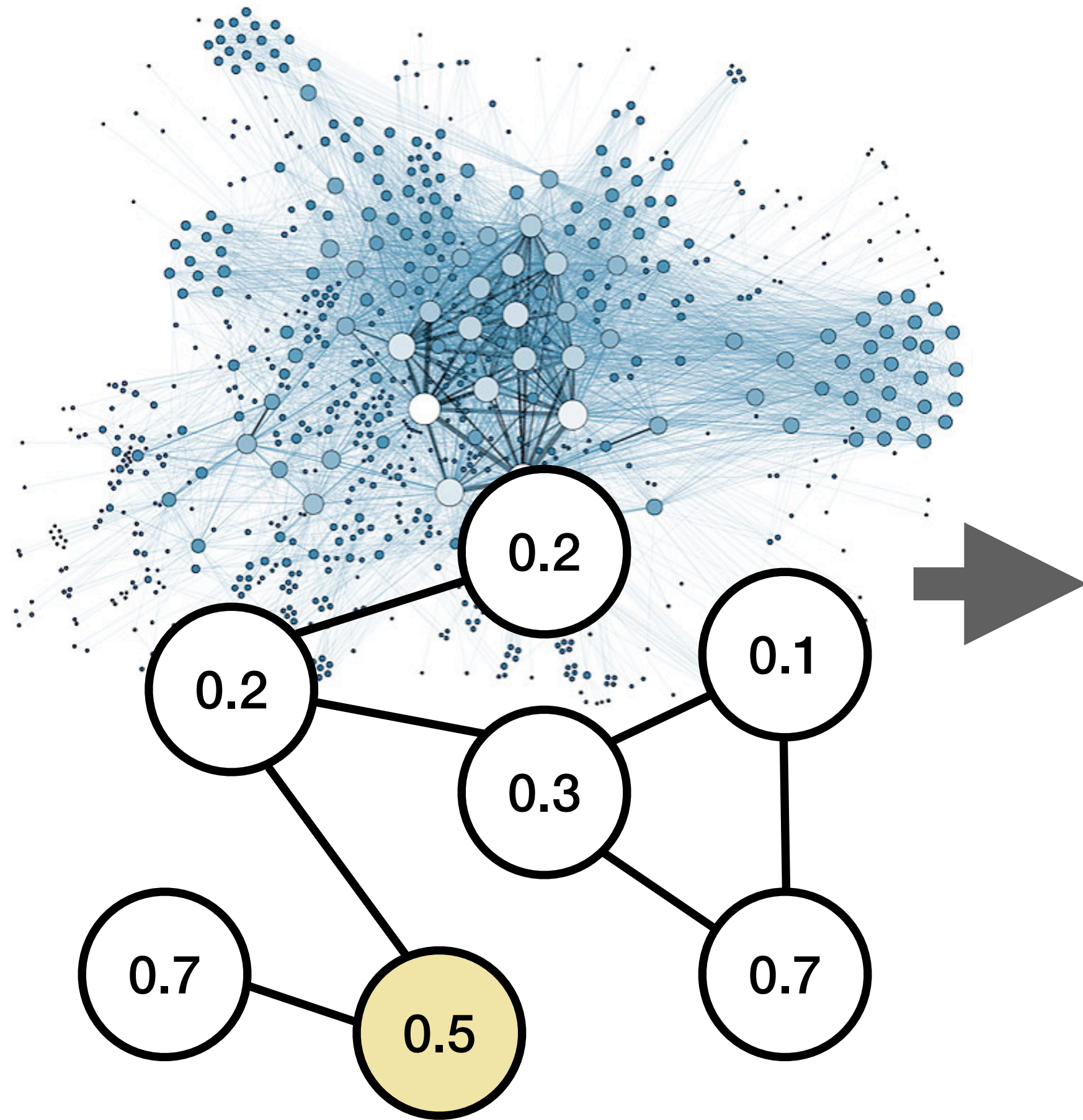
Capas MPNN para clasificar grafos

- Único cambio: **aggregator** al final, toma los embeddings de cada nodo y produce un **embedding único del grafo**.

Training	Value
	1
	0
...	...
...	...
	
	

Capas MPNN para clasificar grafos

- Único cambio: **aggregator** al final, toma los embeddings de cada nodo y produce un **embedding único del grafo**.



$$\sigma \left(\text{Aggregate} \left(\begin{array}{|c|} \hline 0.22452 \\ \hline 0.84369 \\ \hline 0.11241 \\ \hline 0.12459 \\ \hline 0.72629 \\ \hline 0.78121 \\ \hline 0.56291 \\ \hline \dots \\ \hline \end{array} \right) \right) = 0.42851$$

