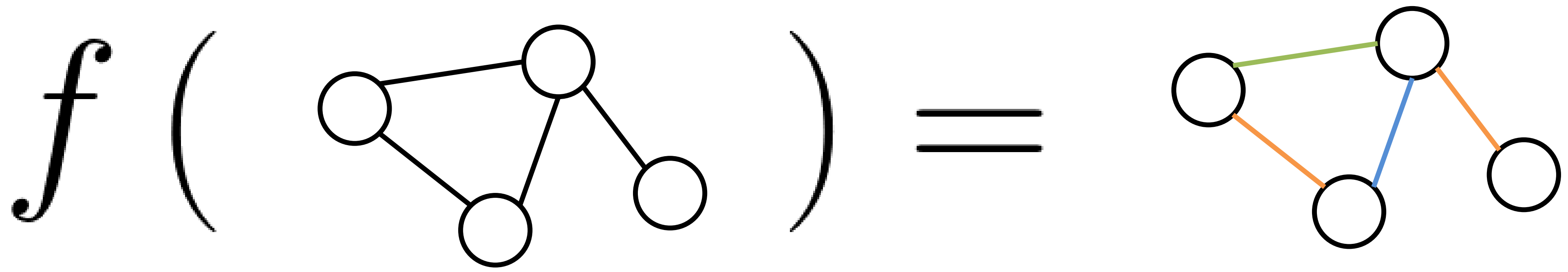
Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ingeniería de Transporte y Logística

Sistemas Urbanos Inteligentes

Redes de grafos convolucionales

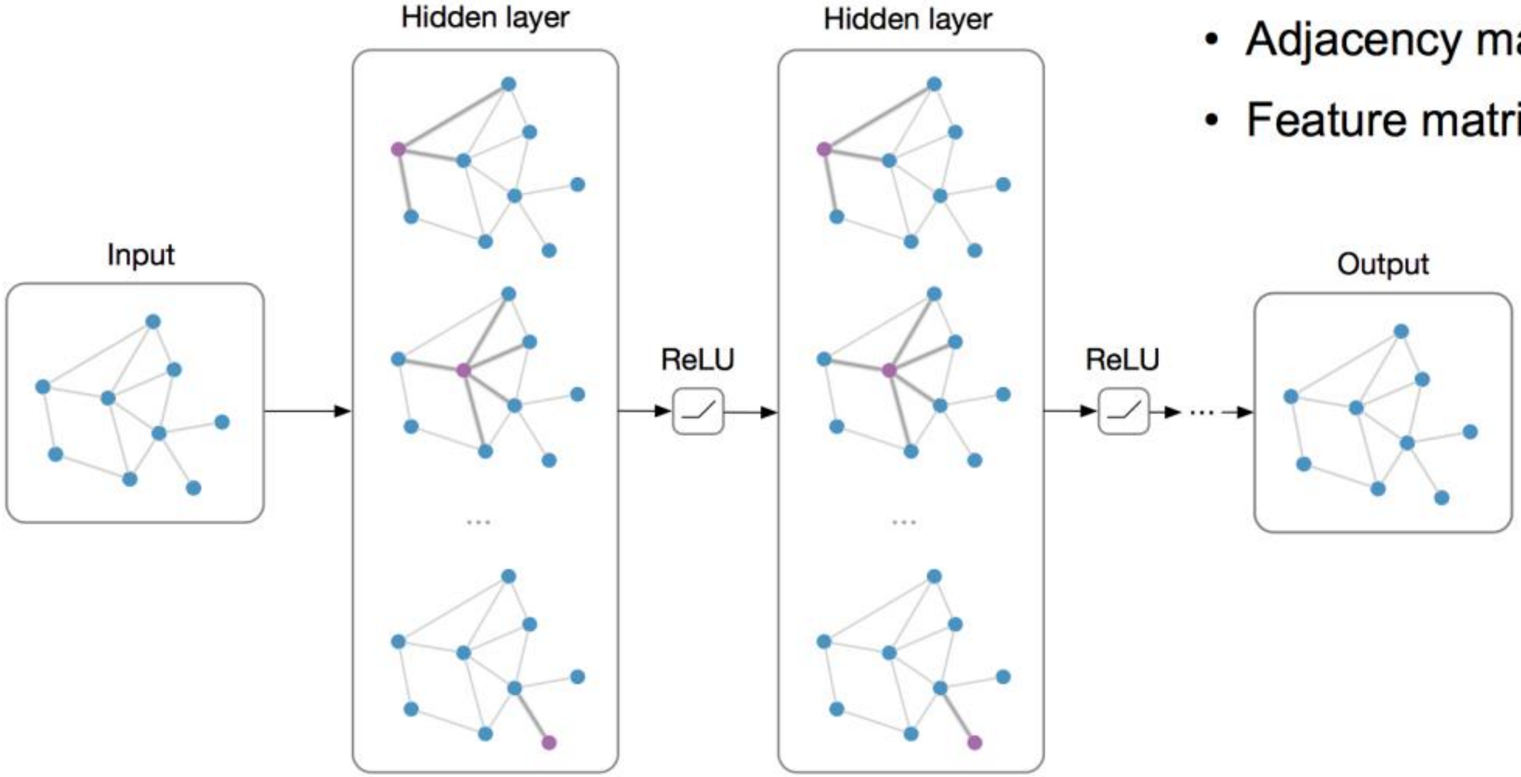
Hans Löbel
Dpto. Ingeniería de Transporte y Logística
Dpto. Ciencia de la Computación

No olvidemos lo que buscamos



Si queremos utilizar redes neuronales para parametrizar f, necesitamos que esta sea differenciable, componible y escalable.

Hacia dónde vamos



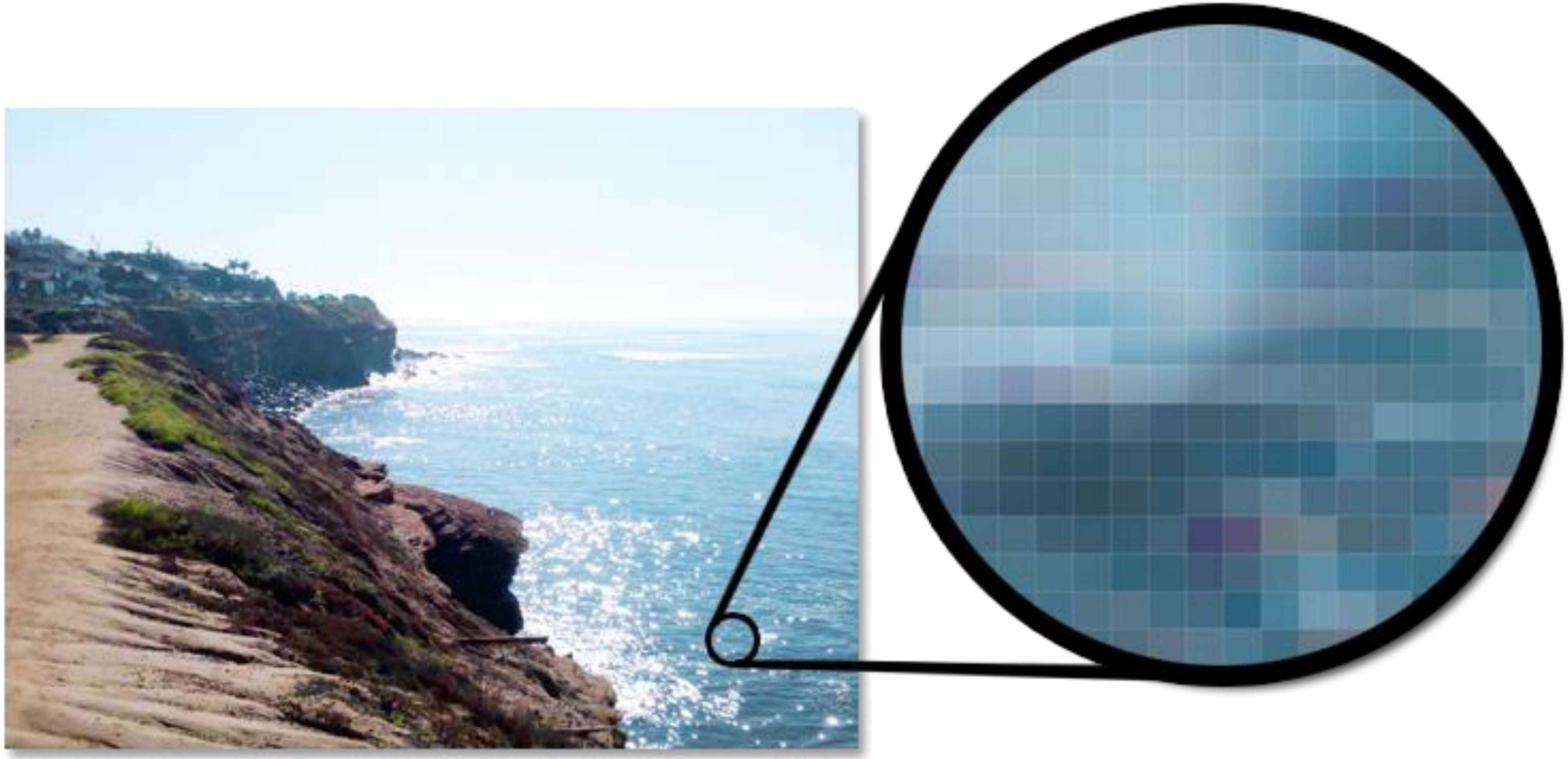
Notation: $\mathcal{G} = (\mathbf{A}, \mathbf{X})$

- Adjacency matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$
- Feature matrix $\mathbf{X} \in \mathbb{R}^{N \times F}$

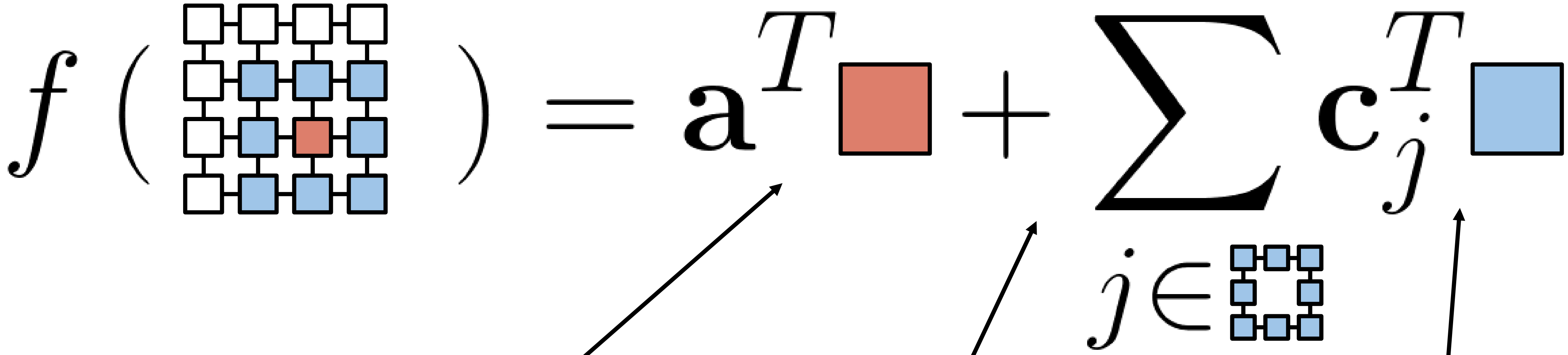
Idea principal: pasar mensajes entre nodos y combinarlos

Otra perspectiva más ML: pasar mensajes entre nodos para refinar la representación

Recordemos la idea detrás de las convoluciones



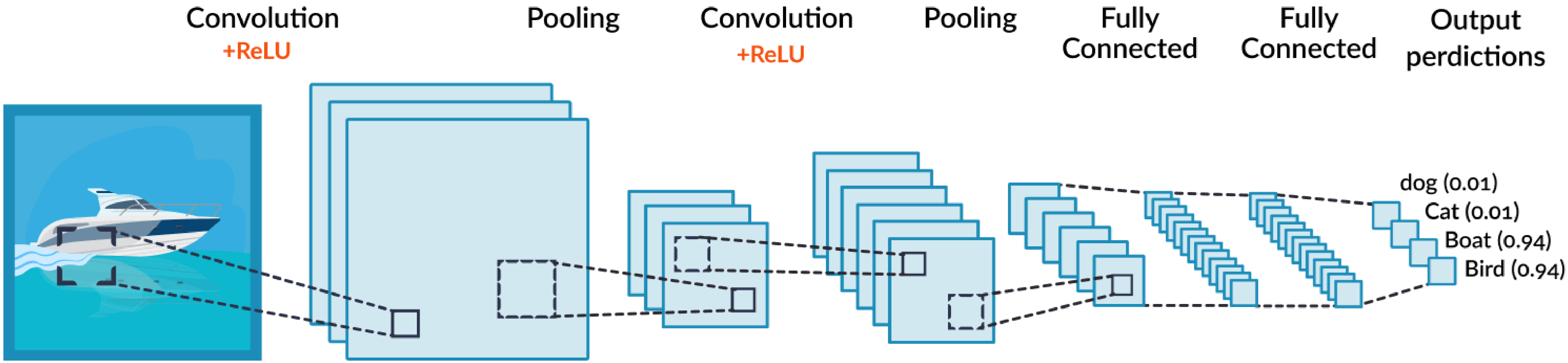
Recordemos la idea detrás de las convoluciones



Operación local
(pixel central)

Agregación

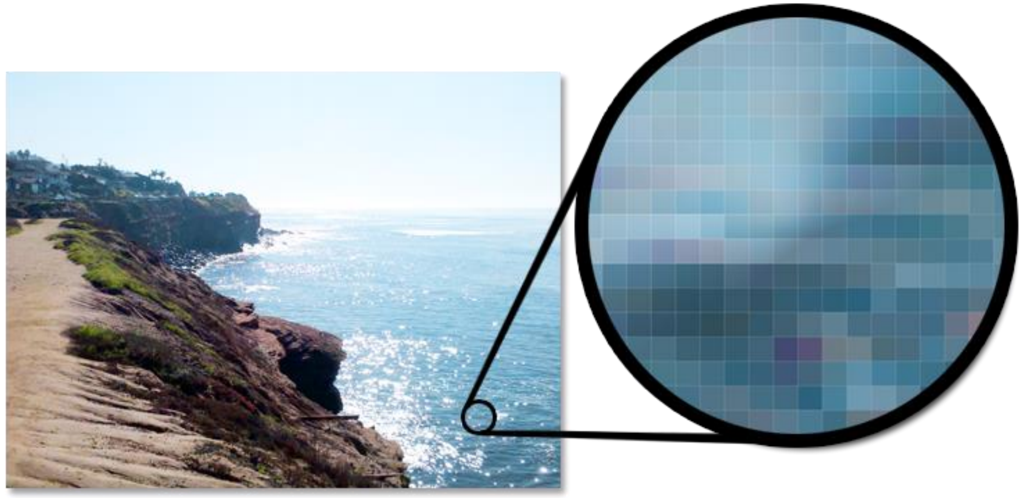
Operación sobre
vecinos (resto de
los pixeles)

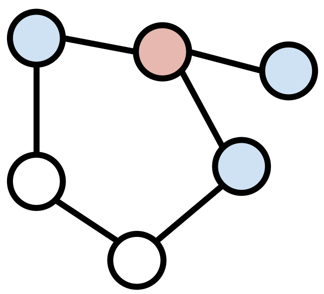

Convolution
+ReLU
Pooling
Convolution
+ReLU
Pooling
Fully
Connected
Fully
Connected
Output
perdictions
dog (0.01)
Cat (0.01)
Boat (0.94)
Bird (0.94)

¿En qué se diferencian las imágenes de los grafos?

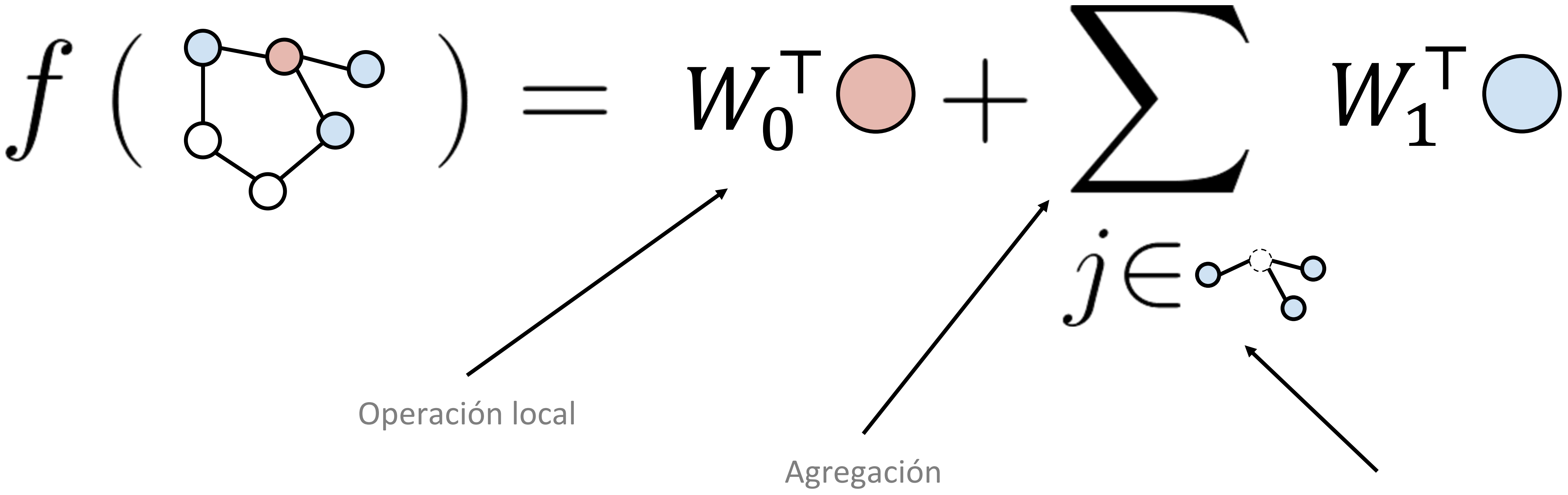
La clave es fijarse en la estructura local

- ✓ Localidad (vecindario)
- ✗ Tamaño del vecinario
- ✗ Orden del vecindario





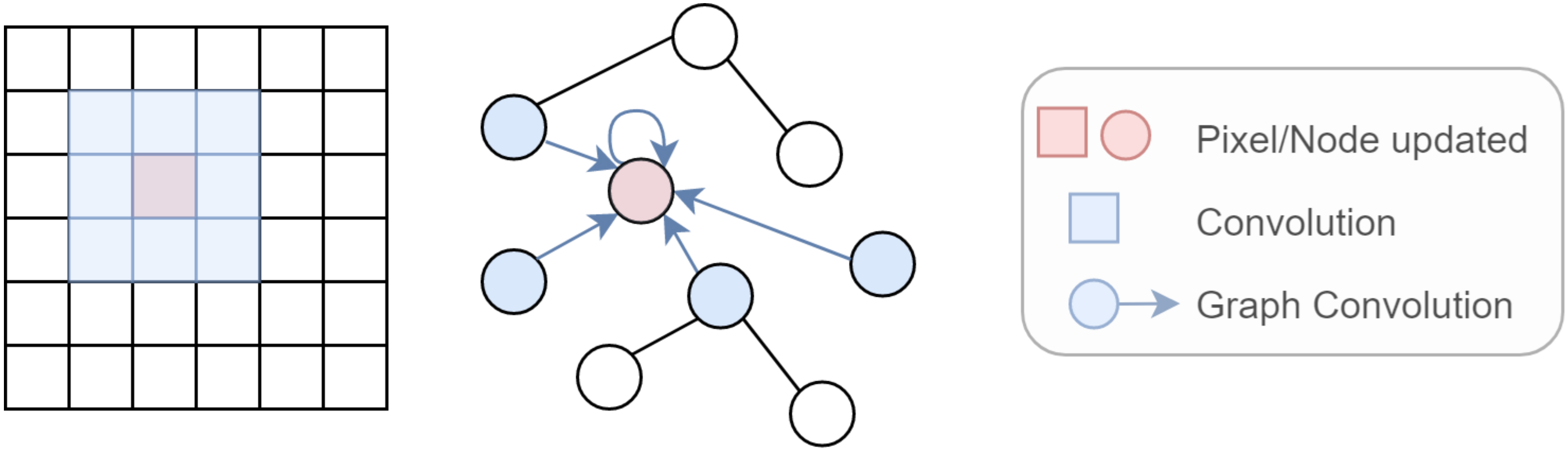
Convoluciones en grafos (versión simplificada)



Operación sobre vecinos usando matriz de adyacencia

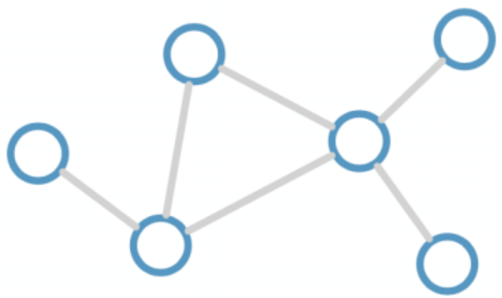
Es importante notar que acá solo podemos aprender una transformación que no depende del orden/posición del vecino

Convoluciones en grafos (versión simplificada)

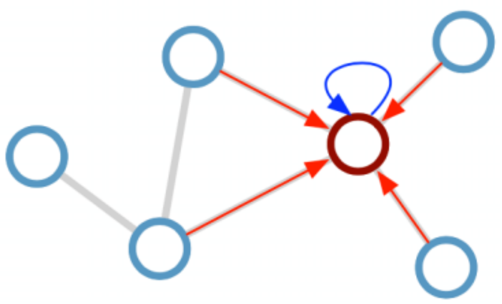


Convoluciones en grafos (un poco menos simplificadas)

Consider this undirected graph:



Calculate update for node in red:



Desirable properties:

- Weight sharing over all locations
- Invariance to permutations
- Linear complexity O(E)
- Applicable both in transductive and inductive settings

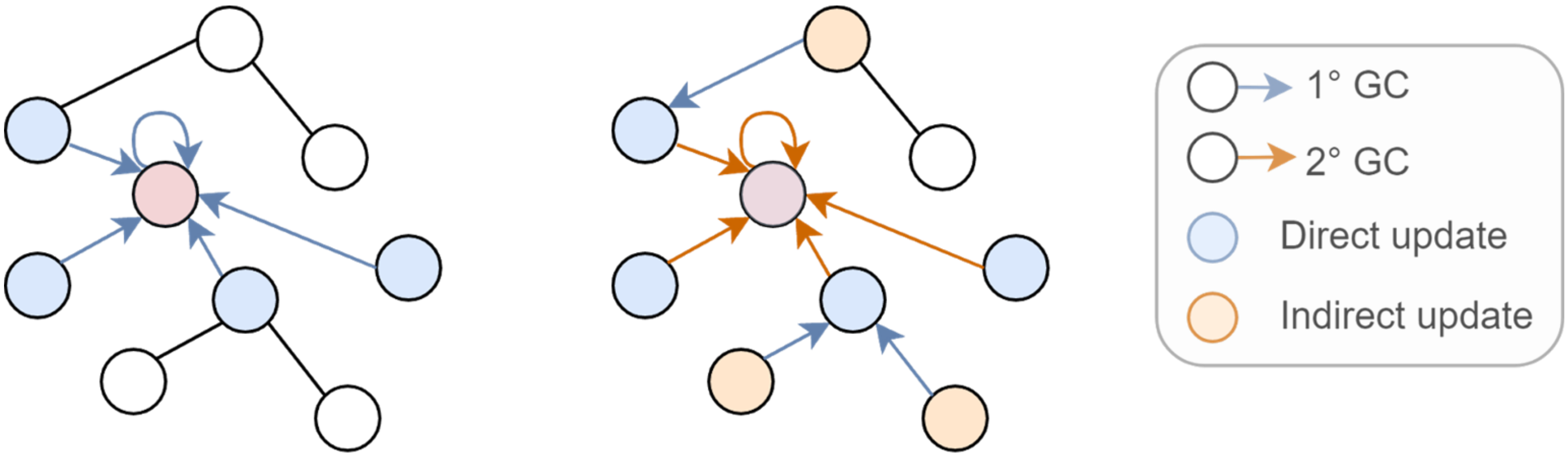
Update rule:

$$\mathbf{h}_i^{(l+1)} = \sigma\left(\mathbf{h}_i^{(l)}\mathbf{W}_0^{(l)} + \sum_{j\in\mathcal{N}_i}\frac{1}{c_{ij}}\mathbf{h}_j^{(l)}\mathbf{W}_1^{(l)}\right)$$

Scalability: subsample messages [Hamilton et al., NIPS 2017]

$\mathcal{N}_i$: neighbor indices c_{ij} : norm. constant

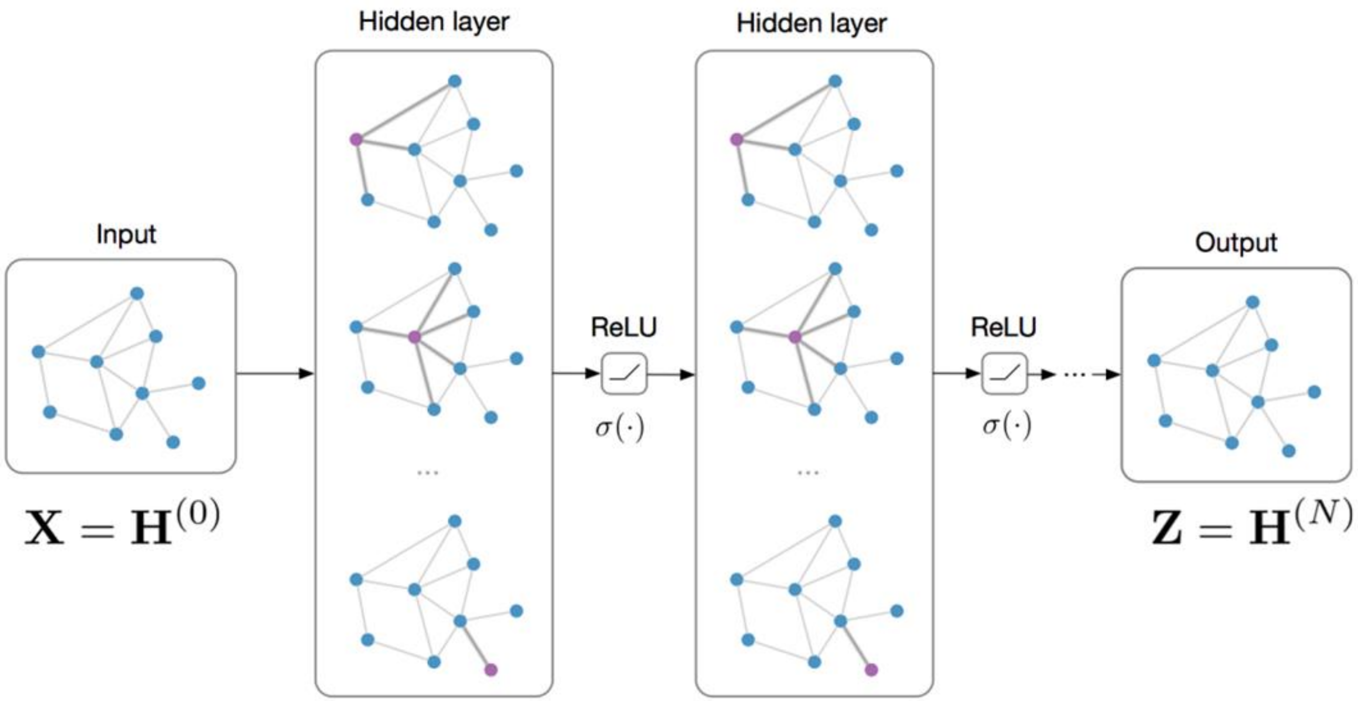
¿Qué pasa si tenemos/queremos más capas?



Capas sucesivas amplían la "visión" del vecindario de cada nodo

¿Cómo se ve en la práctica una GCN?

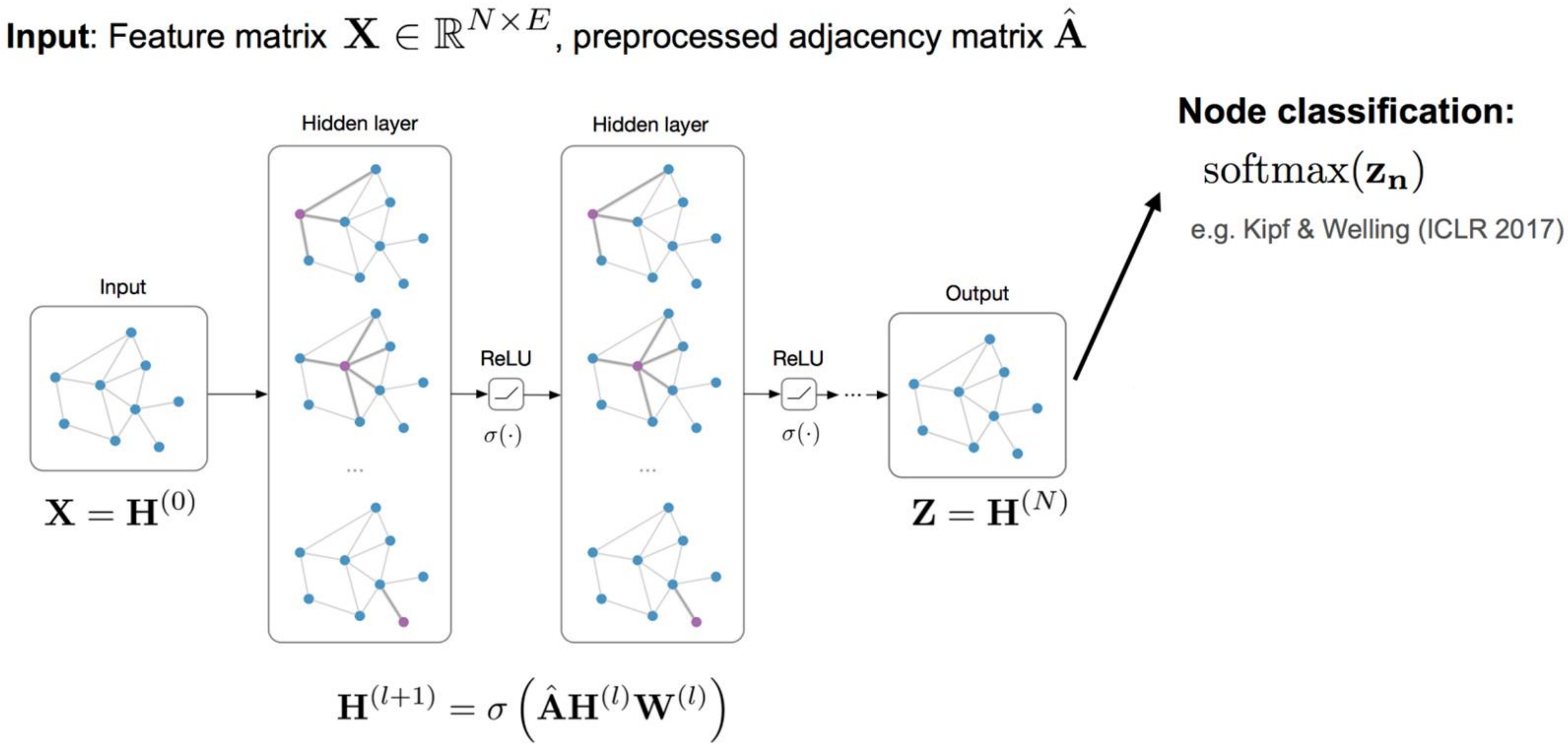
Input: Feature matrix $\mathbf{X} \in \mathbb{R}^{N \times E}$, preprocessed adjacency matrix $\hat{\mathbf{A}}$



$$\mathbf{H}^{(l+1)} = \sigma\left(\hat{\mathbf{A}}\mathbf{H}^{(l)}\mathbf{W}^{(l)}\right)$$

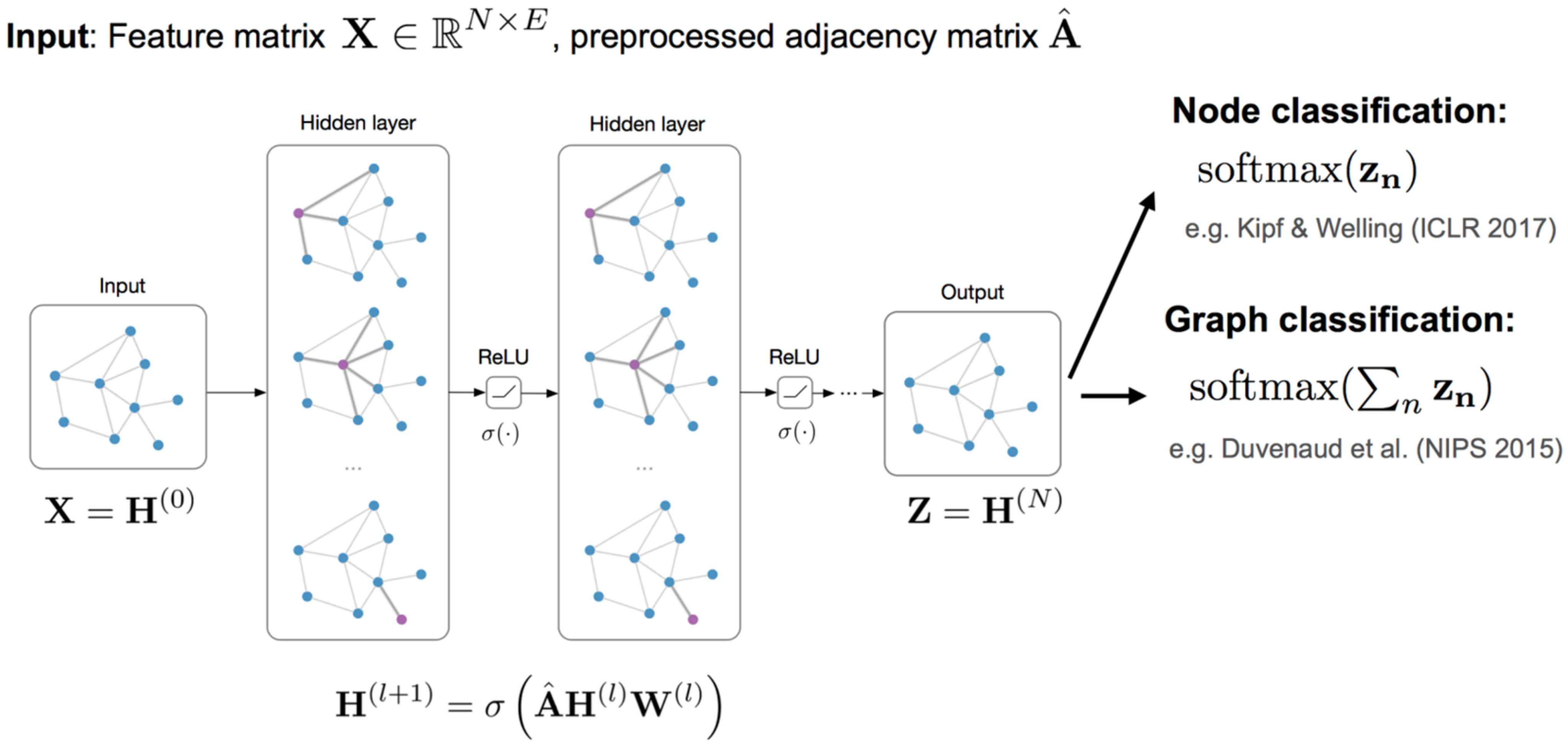
Kipf, T.N. and Welling, M., 2016. Semi-supervised classification with graph convolutional networks.

¿Cómo se ve en la práctica una GCN?



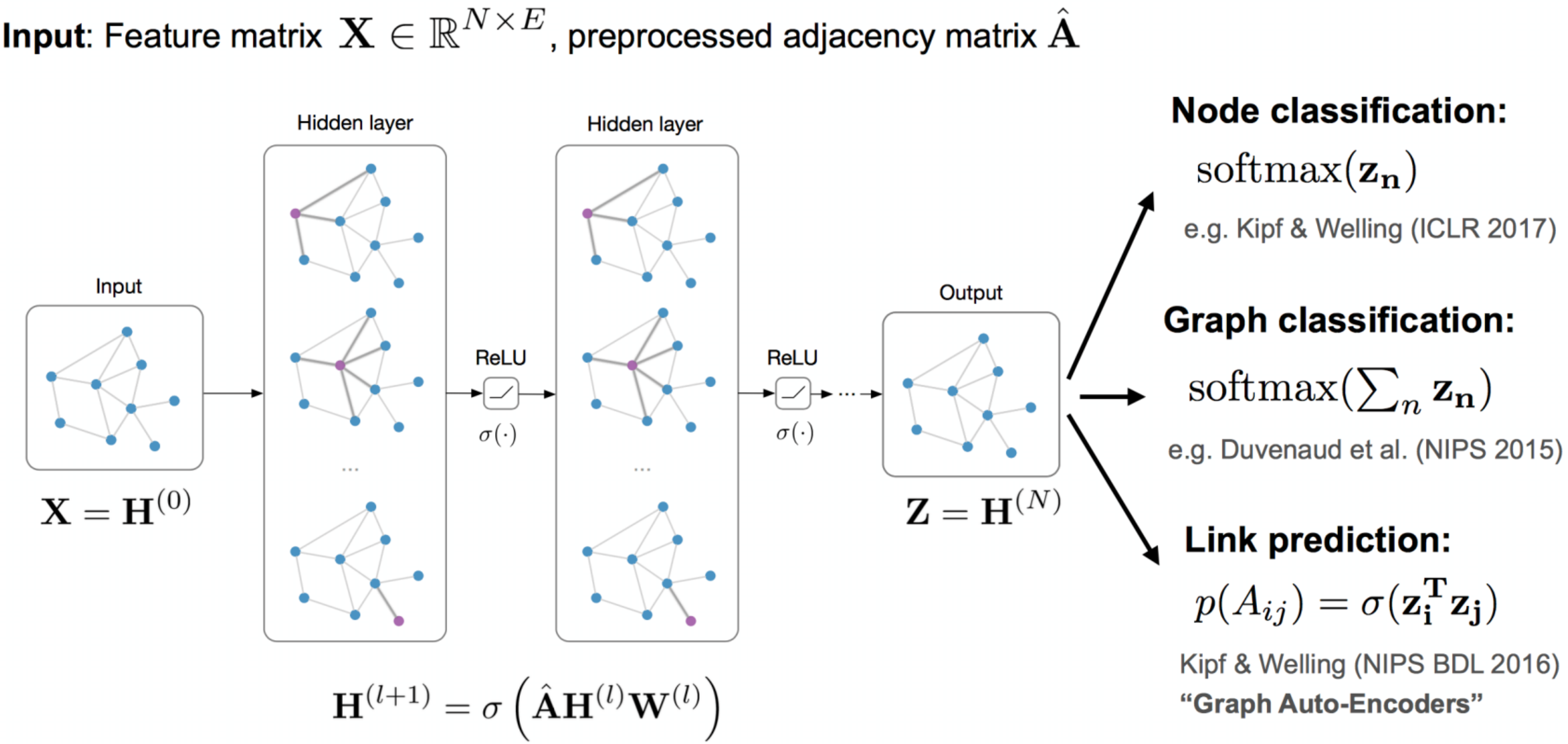
Kipf, T.N. and Welling, M., 2016. Semi-supervised classification with graph convolutional networks.

¿Cómo se ve en la práctica una GCN?



Kipf, T.N. and Welling, M., 2016. Semi-supervised classification with graph convolutional networks.

¿Cómo se ve en la práctica una GCN?



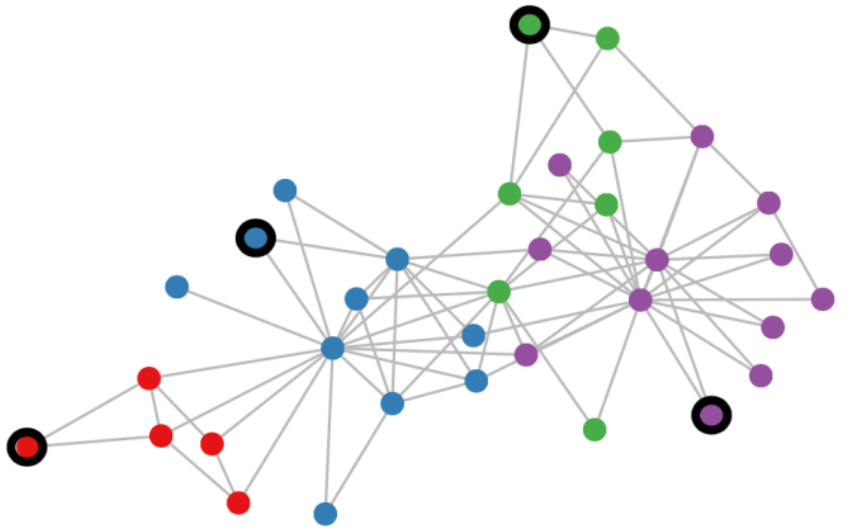
Kipf, T.N. and Welling, M., 2016. Semi-supervised classification with graph convolutional networks.

También podemos hacer clasificación semi-supervisada

Setting:

Some nodes are labeled (black circle)
All other nodes are unlabeled

Task:
Predict node label of unlabeled nodes



Evaluate loss on labeled nodes only:

$$\mathcal{L} = -\sum_{l \in \mathcal{Y}_L} \sum_{f=1}^{F} Y_{lf} \ln Z_{lf}$$

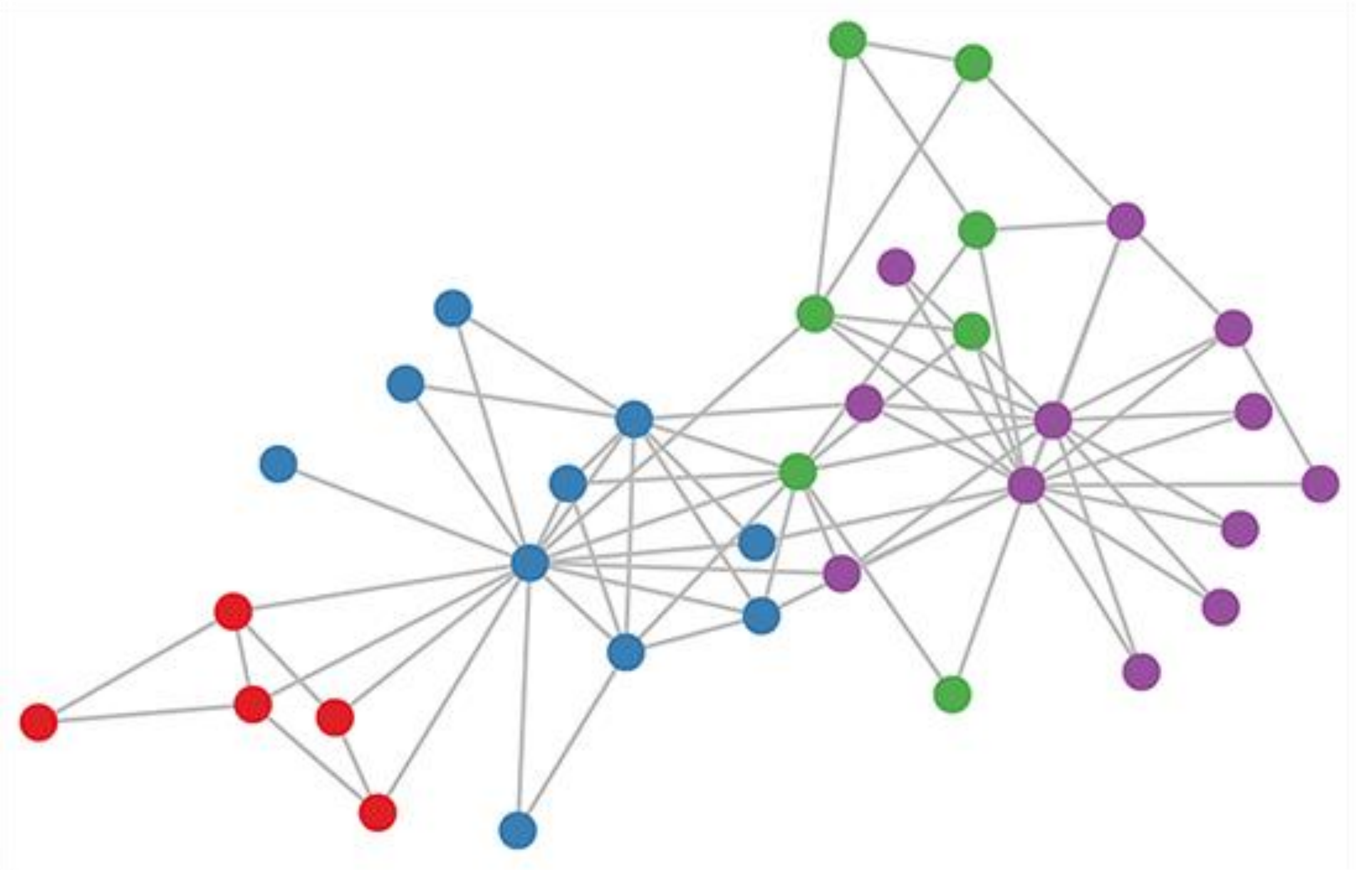
$\mathcal{Y}_L$ set of labeled node indices

$\mathbf{Y}$ label matrix

$\mathbf{Z}$ GCN output (after softmax)

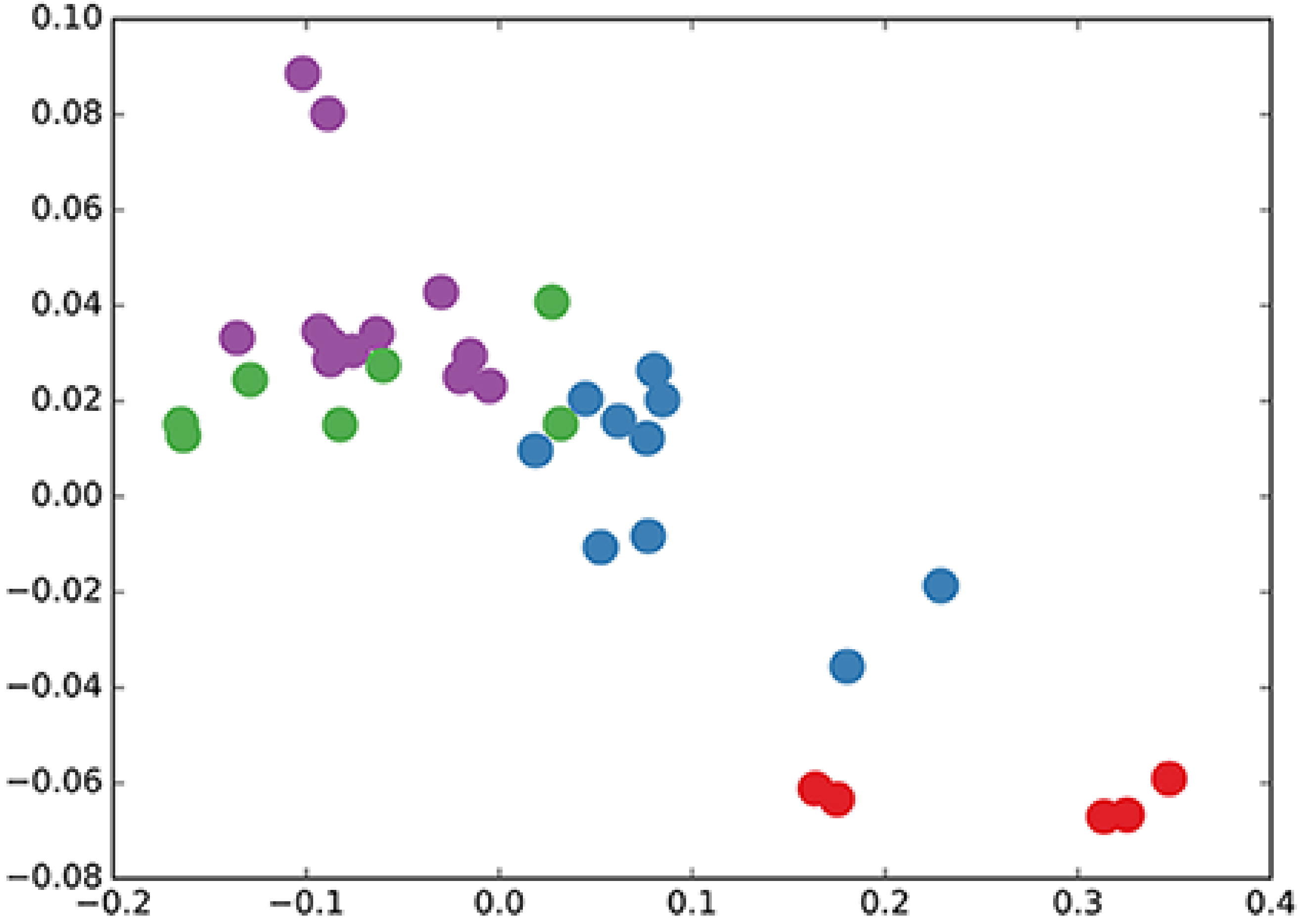
Kipf, T.N. and Welling, M., 2016. Semi-supervised classification with graph convolutional networks.

Veamos como funciona esto



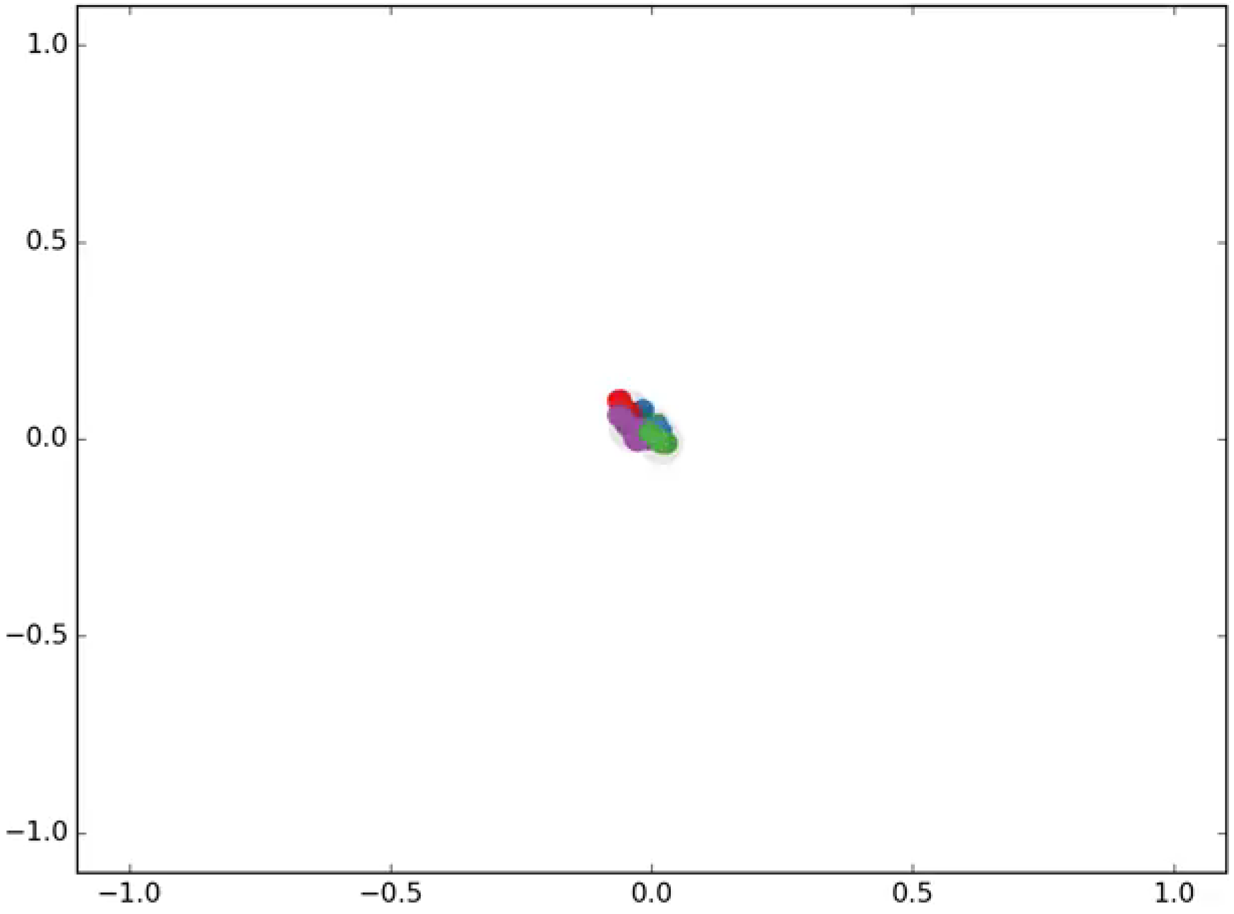
Kipf, T.N. and Welling, M., 2016. Semi-supervised classification with graph convolutional networks.

Veamos como funciona esto



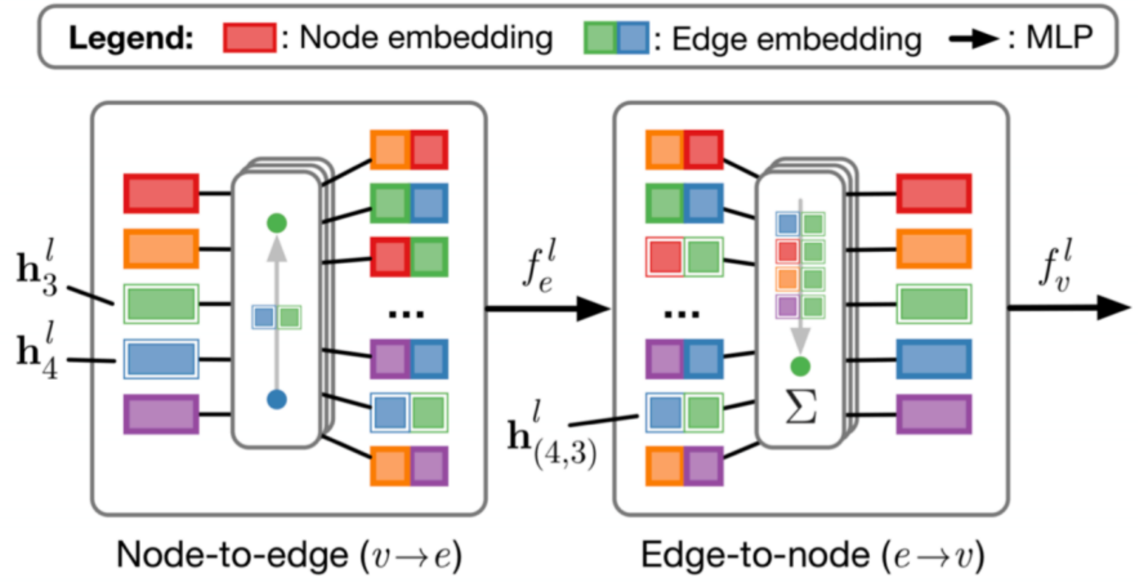
Kipf, T.N. and Welling, M., 2016. Semi-supervised classification with graph convolutional networks.

Veamos como funciona esto



Kipf, T.N. and Welling, M., 2016. Semi-supervised classification with graph convolutional networks.

Podemos extender todo esto para calcular representaciones de arcos



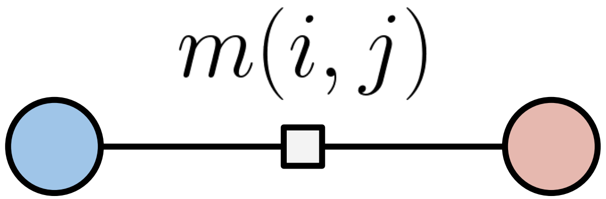
Formally:

$$v \to e: \quad \mathbf{h}_{(i,j)}^l = f_e^l([\mathbf{h}_i^l, \mathbf{h}_j^l, \mathbf{x}_{(i,j)}])$$

$$e \to v: \quad \mathbf{h}_j^{l+1} = f_v^l([\textstyle\sum_{i \in \mathcal{N}_j} \mathbf{h}_{(i,j)}^l, \mathbf{x}_j])$$

El problema es que resulta mucho más complejo

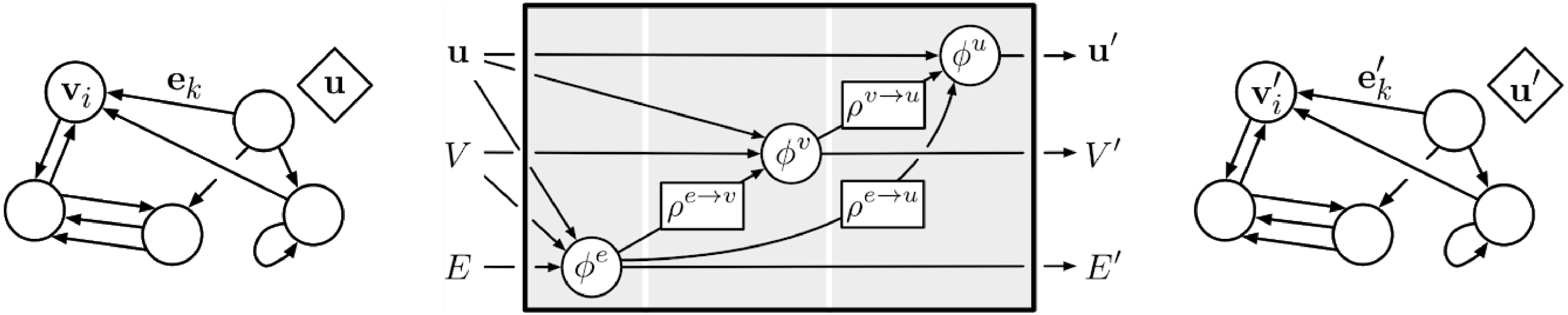
Y también podemos generalizarlo completamente



En vez de utilizar la matriz de adyacencia de forma directa, podamos pensar que los nodos intercambian mensajes de acuerdo a diversos mecanismos, por ejemplo, atención.

Gilmer, J., Schoenholz, S.S., Riley, P.F., Vinyals, O. and Dahl, G.E., 2017, July. Neural message passing for quantum chemistry.

Y también podemos generalizarlo completamente



Battaglia, P.W., Hamrick, J.B., Bapst, V., Sanchez-Gonzalez, A., Zambaldi, V., Malinowski, M., Tacchetti, A., Raposo, D., Santoro, A., Faulkner, R. and Gulcehre, C., 2018. Relational inductive biases, deep learning, and graph networks

Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ingeniería de Transporte y Logística

Sistemas Urbanos Inteligentes

Redes de grafos convolucionales

Hans Löbel
Dpto. Ingeniería de Transporte y Logística
Dpto. Ciencia de la Computación