Graph attention networks

Sistemas urbanos inteligentes

Felipe Gutiérrez

TL:DR

- Incorporación de mecanismos de atención en redes de grafo.
- Testeo en datasets apropiados para benchmark.
- Se alcanzaron resultados de estado de arte.

Outline

- Contexto
- Atención
- Redes de grafo
- GAT
- Benchmarks

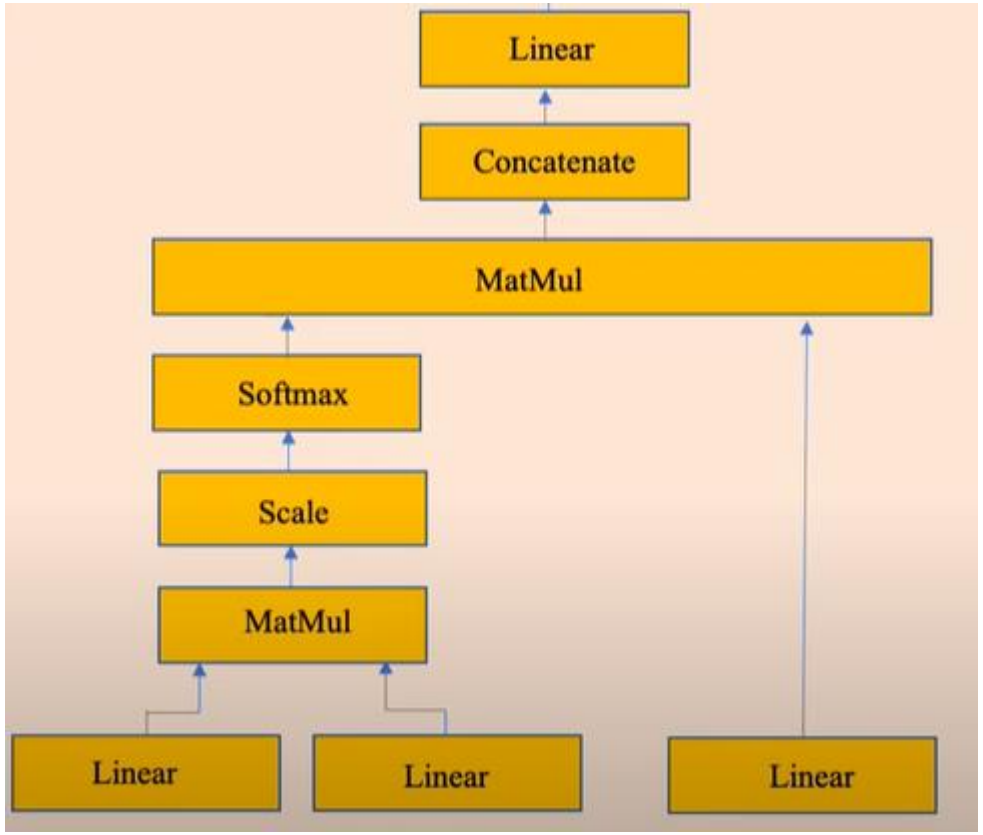
Contexto

- Estructuras de datos “Euclidianas”
  - Imágenes
  - Tablas
  - Textos
  - Etc
- Estructuras no euclideanas
  - Redes sociales
  - Mallas 3D
  - Redes telecomunicación
  - Etc

Contexto

- GNN
  - Técnicas de dl adaptadas a grafos
  - “Learning a suitable representation of graph data”(DeepFindr)
- Métodos espectrales
  - Poco adaptables a variaciones de grafos
  - Alto costo computacional
- Métodos no espectrales
  - Generalización de convoluciones (MoNet)
- Mecanismos de atención
  - DeFacto actual para datos secuenciales

Atención



- Input: embedding de secuencia.
- Capas lineales (Query Q, Key K, Value V)
- Similaridad (Query y Key)
- Dot product (Anterior y Value)
- Concatenación (multihead)
- Linear

NOTAR: $\text{Cos}\,(A, B) = \frac{A \,.\, B}{|A|\,|B|}$

$$\text{Attention(Q, K, V)} = \text{softmax}\left(\frac{Q.K^T}{\sqrt{d_k}}\right).V$$

“Dada una Query cuales son los elementos importantes de la llave para obtener el valor”

GNN

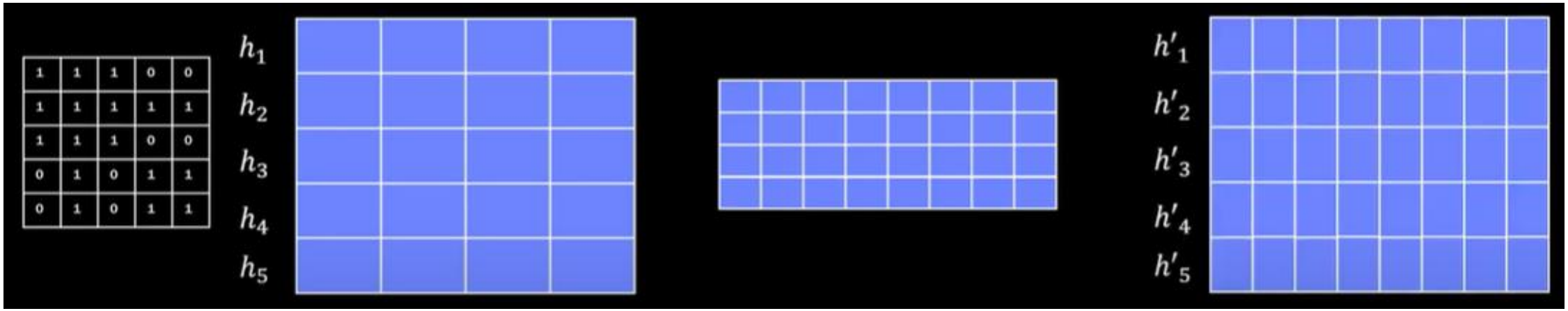
$$h_i^{l+1} = \sigma(\sum_{j \in N(i)} W * h_j)$$

$$A = h \times h,\ A_h \in Z/2Z^h$$

$$F = h \times f,\ F_h \in R^f$$

$$W = f \times w,\ W_f \in R^w$$

$$E = \sigma(A * F * W)$$



Nota: La diagonal contiene unos

GAN

¿Que tan importante es un nodo j para el nodo i?

$$e_{ij} = a(\mathbf{W}\vec{h}_i, \mathbf{W}\vec{h}_j)$$

$$\alpha_{ij} = \mathrm{softmax}_j(e_{ij}) = \frac{\exp(e_{ij})}{\sum_{k \in \mathcal{N}_i} \exp(e_{ik})}.$$

Gan

- ¿Qué tan importante es el nodo j para el nodo i?
- Se introduce coeficiente de atención y se normaliza.
- Generalmente:

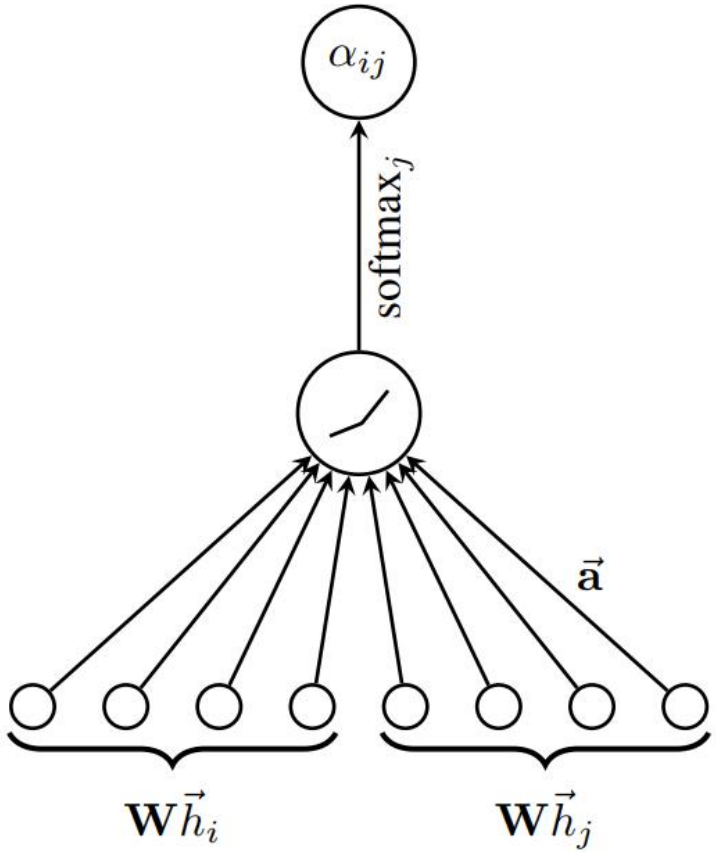
$$e_{ij} = a(\mathbf{W}\vec{h}_i, \mathbf{W}\vec{h}_j)$$

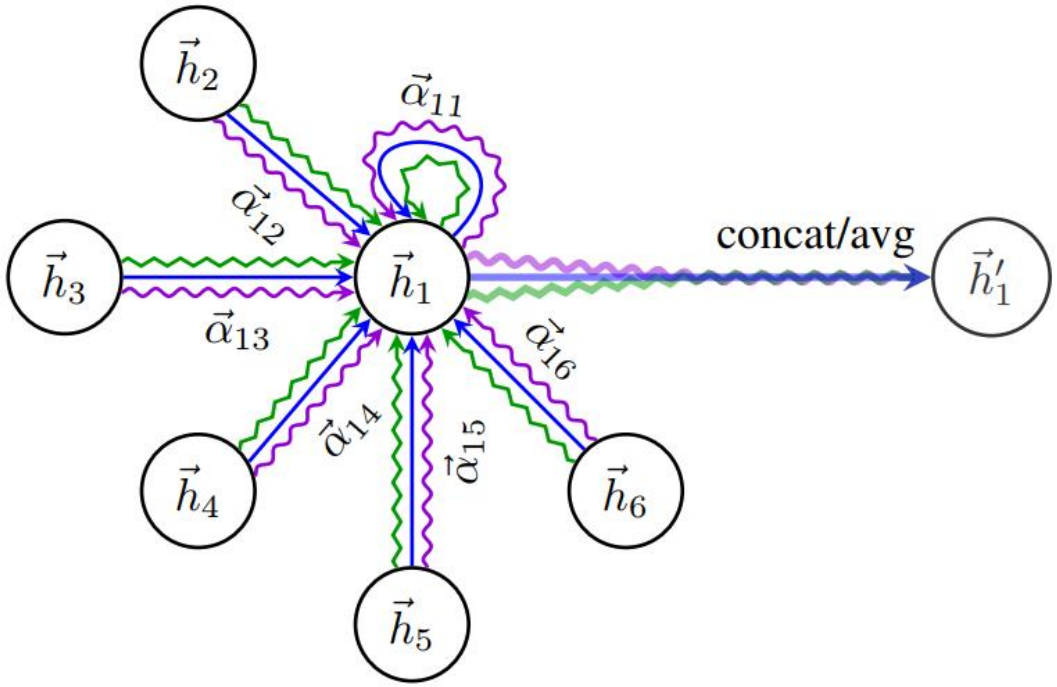
$$\alpha_{ij} = \mathrm{softmax}_j(e_{ij}) = \frac{\exp(e_{ij})}{\sum_{k\in\mathcal{N}_i}\exp(e_{ik})}.$$

- En paper:

$$\alpha_{ij} = \frac{\exp\left(\mathrm{LeakyReLU}\left(\vec{\mathbf{a}}^T[\mathbf{W}\vec{h}_i \| \mathbf{W}\vec{h}_j]\right)\right)}{\sum_{k\in\mathcal{N}_i}\exp\left(\mathrm{LeakyReLU}\left(\vec{\mathbf{a}}^T[\mathbf{W}\vec{h}_i \| \mathbf{W}\vec{h}_k]\right)\right)}$$

GAN

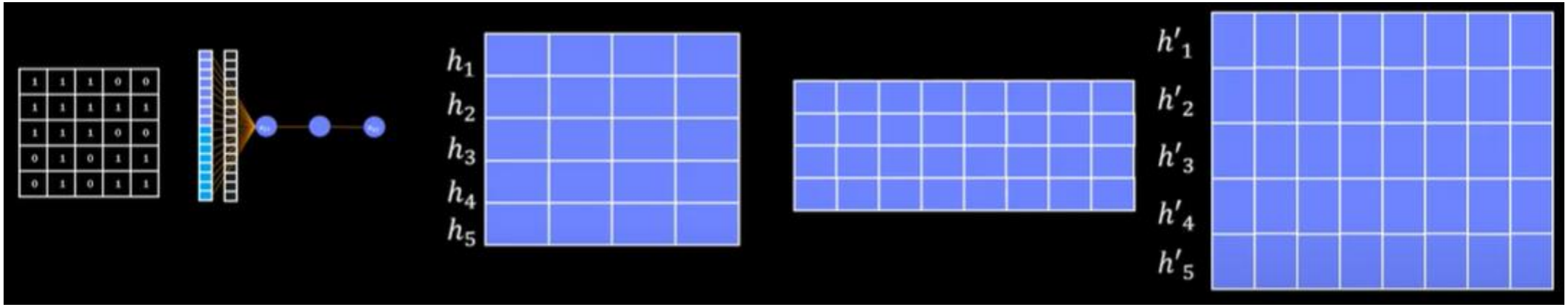

α_{ij}
softmax$_j$
$\vec{\mathbf{a}}$
$\mathbf{W}\vec{h}_i$
$\mathbf{W}\vec{h}_j$


$\vec{h}_2$
$\vec{\alpha}_{11}$
$\vec{\alpha}_{12}$
concat/avg
$\vec{h}_3$
$\vec{h}_1$
$\vec{h}'_1$
$\vec{\alpha}_{13}$
$\vec{\alpha}_{16}$
$\vec{\alpha}_{14}$
$\vec{\alpha}_{15}$
$\vec{h}_4$
$\vec{h}_6$
$\vec{h}_5$

Gan

- Finalmente, podemos interpretar la incorporación del mecanismo de atención como una ponderación de la matriz de adyacencia

$$h_i^{l+1} = \sigma(\sum_{j \in N(i)} \alpha_{ij} W * h_j)$$



Ventajas

- Paralelizable
- Reducción de operaciones costosas
- Dar una noción de importancia a distintos nodos de una vecindad
- Aplicable a grafos dirigidos
- No asume ningún orden previo
- Solo depende de características de cada nodo y no propiedades estructurales

Evaluación

- Datasets:
  - Cora – Clasificación de nodos, red de citación
  - Citeseer – Clasificación de nodos, red de citación
  - Pubmed – Clasificación de nodos, red de citación
  - PPI – Interacción entre proteínas

Evaluación

Table 1: Summary of the datasets used in our experiments.

	Cora	Citeseer	Pubmed	PPI
Task	Transductive	Transductive	Transductive	Inductive
# Nodes	2708 (1 graph)	3327 (1 graph)	19717 (1 graph)	56944 (24 graphs)
# Edges	5429	4732	44338	818716
# Features/Node	1433	3703	500	50
# Classes	7	6	3	121 (multilabel)
# Training Nodes	140	120	60	44906 (20 graphs)
# Validation Nodes	500	500	500	6514 (2 graphs)
# Test Nodes	1000	1000	1000	5524 (2 graphs)

Transductive

Method	**Cora**	**Citeseer**	**Pubmed**
MLP	55.1%	46.5%	71.4%
ManiReg (Belkin et al., 2006)	59.5%	60.1%	70.7%
SemiEmb (Weston et al., 2012)	59.0%	59.6%	71.7%
LP (Zhu et al., 2003)	68.0%	45.3%	63.0%
DeepWalk (Perozzi et al., 2014)	67.2%	43.2%	65.3%
ICA (Lu & Getoor, 2003)	75.1%	69.1%	73.9%
Planetoid (Yang et al., 2016)	75.7%	64.7%	77.2%
Chebyshev (Defferrard et al., 2016)	81.2%	69.8%	74.4%
GCN (Kipf & Welling, 2017)	81.5%	70.3%	**79.0%**
MoNet (Monti et al., 2016)	81.7 ± 0.5%	—	78.8 ± 0.3%
GCN-64*	81.4 ± 0.5%	70.9 ± 0.5%	**79.0** ± 0.3%
GAT (ours)	**83.0** ± 0.7%	**72.5** ± 0.7%	**79.0** ± 0.3%

Inductive

Method	**PPI**
Random	0.396
MLP	0.422
GraphSAGE-GCN (Hamilton et al., 2017)	0.500
GraphSAGE-mean (Hamilton et al., 2017)	0.598
GraphSAGE-LSTM (Hamilton et al., 2017)	0.612
GraphSAGE-pool (Hamilton et al., 2017)	0.600
GraphSAGE*	0.768
Const-GAT (ours)	0.934 ± 0.006
GAT (ours)	**0.973** ± 0.002

Visualización Cora

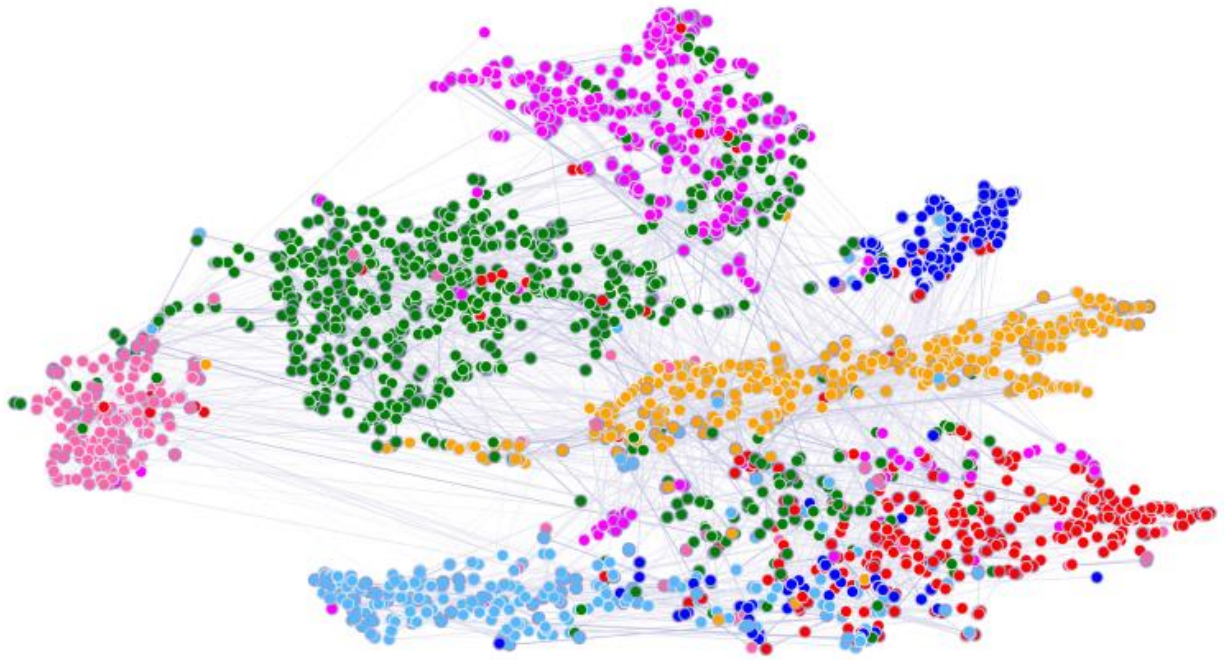


Figure 2: A t-SNE plot of the computed feature representations of a pre-trained GAT model's first hidden layer on the Cora dataset. Node colors denote classes. Edge thickness indicates aggregated normalized attention coefficients between nodes i and j, across all eight attention heads $(\sum_{k=1}^{K} \alpha_{ij}^{k} + \alpha_{ji}^{k})$.

Conclusiones

- Reiterar lo dicho en ventajas
- Atiende a problemas teóricos de métodos espectrales
- Logro resultados de estado del arte
- Trabajos futuros necesarios en:
  - Manejo de batches
  - Clasificación de grafos (hasta ahora acotada a nodos)
  - Incorporación de features de enlaces

Creditos/Agradecimientos

- Velickovic et al
  - Graph Attention Networks (2018)

- DeepFindr
  - Understanding Graph Attention Networks
  - https://www.youtube.com/watch?v=A-yKQamf2Fc

- Hedu – Math of Intelligence
  - Visual Guide to Transformer Neural Network
  - https://www.youtube.com/watch?v=mMa2PmYJlCo