Graph Neural Networks summary

E. Eleftheriadis, Prof. Ioannis Pitas
Aristotle University of Thessaloniki
pitas@csd.auth.gr
www.aiia.csd.auth.gr
Version 2.7

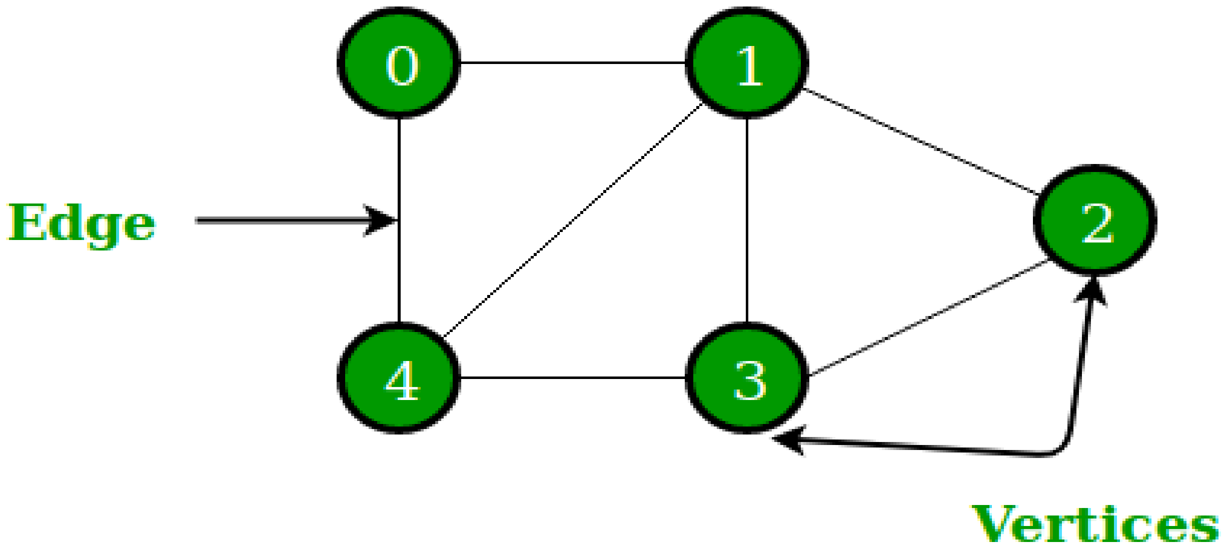
Artificial Intelligence & Information Analysis Lab

Graph Neural Networks

- **Introduction to Graphs**
- Neural Networks
- Graph Convolutional Networks (GCN)
- Recurrent Graph Neural Networks (RGNN)
- Graph Auto-Encoders
- Spatial-Temporal Graph Neural Networks
- GNN Applications

Introduction to Graphs

- **Graph** is a mathematical data structure that consists of **nodes/vertices** which are linked together with **edges**.



Mathematical Notation

- The mathematical notation of a **graph** $\boldsymbol{G}$ is $\boldsymbol{G} = (\boldsymbol{V}, \boldsymbol{E}, \boldsymbol{W})$ with:
  - $\boldsymbol{V}$ is a set of **vertices** or **nodes**.
  - $\boldsymbol{E} \subseteq \boldsymbol{V} \times \boldsymbol{V}$ is a set of **edges**.
  - $\boldsymbol{W}$ is the **weights** of the edges.

- Two vertices $\boldsymbol{u} \in \boldsymbol{V}$ and $\boldsymbol{v} \in \boldsymbol{V}$ are connected with a edge, if $(\boldsymbol{u}, \boldsymbol{v}) \in \boldsymbol{E}$ or $(\boldsymbol{v},\ \boldsymbol{u}) \in \boldsymbol{E}$, if the graph $\boldsymbol{G}$ is **undirected**.

Graph Representation

- There are two commonly representations of a **graph *G***:
  - **Adjacency matrix**
  - **Adjacency list**

- **Adjacency matrix** is a square matrix that has the value 1 in the index ***A[i,j]*** and ***A[j,i]*** if the i and j nodes are connected.
- **Adjacency list** is an array that consist of all the vertices of the graph and for each index of the array begin a linked list that represent the nodes that are connected with the node in the array.

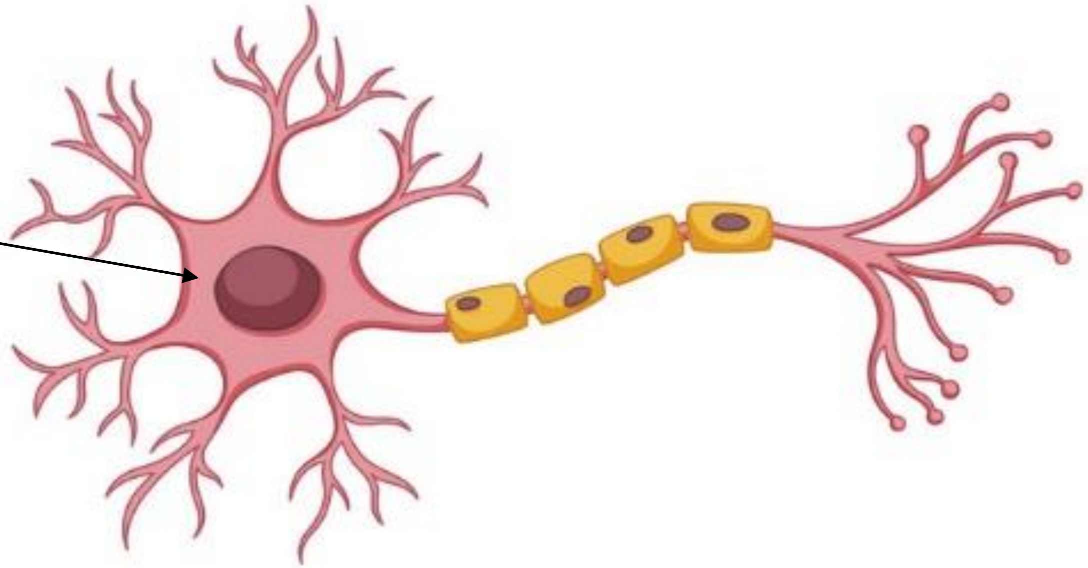
Graph Neural Networks

- Introduction to Graphs
- **Neural Networks**
- Graph Convolutional Networks (GCN)
- Recurrent Graph Neural Networks (RGNN)
- Graph Auto-Encoders
- Spatial-Temporal Graph Neural Networks
- GNN Applications

Artificial Neural Networks

- Artificial Neural Networks mimic human brain to make computations with the data. They have similar structure with the brain:

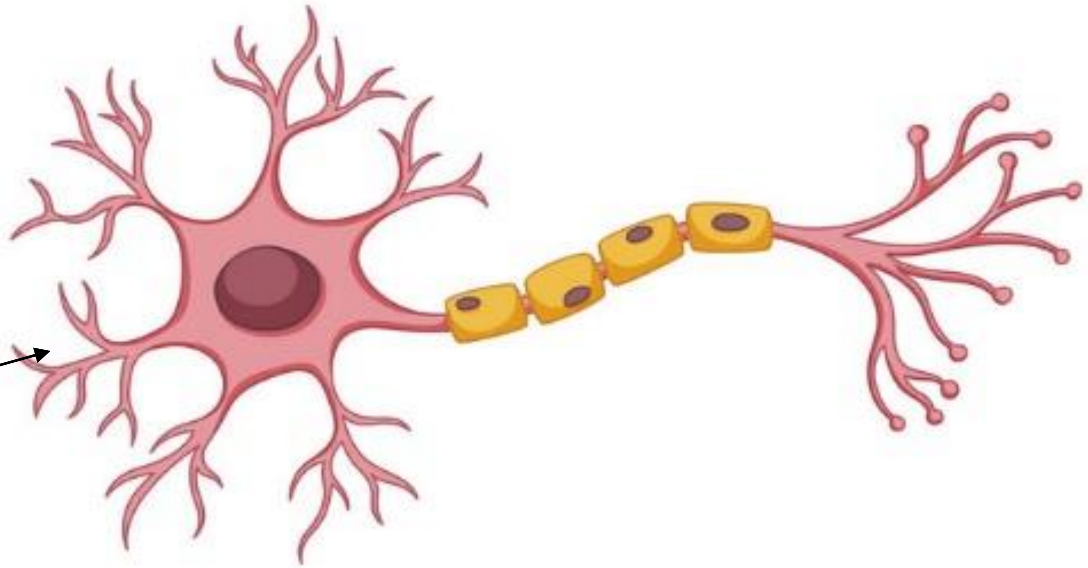
  - **Cell Body**
  - Axon
  - Synapse
  - Dendrite



Artificial Neural Networks

- Artificial Neural Networks mimic human brain to make computations with the data. They have similar structure with the brain:

  - Cell Body
  - Axon
  - Synapse
  - **Dendrite**



Artificial Intelligence & Information Analysis Lab

Artificial Neural Networks

- Input signals: **x** = $[x_1, x_2, \ldots, x_n]^T,\ x_i \in \mathbb{R}$.
- Synaptic weights: **w** = $[w_1, w_2, \ldots, w_n]^T,\quad w_i \in \mathbb{R}$.
- Bias: b, $\quad b \in \mathbb{R}$.
- The input signal with the weights integrate and produce the output signal: **z** = $\left(\sum_{i=1}^{N} w_i x_i\right) + b \ = \ \boldsymbol{w}^T \boldsymbol{x} + b$.

- The **Artificial Neural Networks** have a amount of layers and for each layer exists many *artificial neurons*, that they have to learn a function, as long as the network was training.

$$\widehat{\boldsymbol{y}} = \boldsymbol{f}(\boldsymbol{x}; \boldsymbol{\theta})$$

Graph Neural Networks

- Introduction to Graphs
- Neural Networks
- **Graph Convolutional Networks (GCN)**
- Recurrent Graph Neural Networks (RGNN)
- Graph Auto-Encoders
- Spatial-Temporal Graph Neural Networks
- GNN Applications

Graph Shift Operator

- **Graph Shifted Operation(GSO):** $S \in \mathbb{R}^{N \times N}$, where S is one of the following representation:
  - **Adjacency matrix**
  - **Normalized Adjacency**
  - **Laplacian matrix**
  - **Normalized Laplacian**
- If the adjacency matrix is **symmetric**, then the graph shift operator S is symmetric too ($S^T = S$).

Artificial Intelligence & Information Analysis Lab

Graph Convolution

- The classical convolution of signals can be visualized as a convolution of graph structures.
- We need a **filter** h_k, an **input signal** x_{in} and a **graph shift operator** S.
- The output is:

$$\boldsymbol{Z} = h_0 S^0 \boldsymbol{x_{in}} + h_1 S^1 \boldsymbol{x_{in}} + \ ... + h_k S^k \boldsymbol{x_{in}}$$

$$= \sum_{k=0}^{K-1} h_k S^k \boldsymbol{x_{in}}, \text{where } \mathrm{S}^{\mathrm{i}} \text{ is a shift by i.}$$

Artificial Intelligence & Information Analysis Lab

Graph Perceptron

- In the same way with a Machine Learning task, in **Graph Convolution Neural Networks** we try to minimize the error of the output depending the input signals.

$\boldsymbol{x}$

$$\boldsymbol{z} = \sum_{k=0}^{K-1} h_k S^k \boldsymbol{x_{in}}$$

$\boldsymbol{z} = \boldsymbol{\Phi}(\boldsymbol{x}; \boldsymbol{S}, \boldsymbol{h})$

Graph Convolution Networks

- A GNN is a combination of many graph perceptrons.

$$\mathbf{z}_1 = \sum_{k=0}^{K-1} h_{1k} S^k \mathbf{x}_{in} \xrightarrow{\mathbf{z}_1} \mathbf{x}_1 = \sigma[\mathbf{z}_1]$$

x_1

x_1

$$\mathbf{z}_2 = \sum_{k=0}^{K-1} \mathbf{h}_{2k} S^k \mathbf{x}_1 \xrightarrow{\mathbf{z}_2} \mathbf{x}_2 = \sigma[\mathbf{z}_2]$$

Graph Convolution Networks

- There are two different types of graph convolutional networks:

  - **Spectral Graph Convolutional Networks**
  - **Spatial Graph Convolutional Networks**

Spectral Graph Convolution

- The **features** and the **attributes** of the nodes in the graph converted to **signals**.

- The **convolutions** calculated by factoring the **Laplacian matrix** into the eigenvalues and eigenvectors.

Spatial Graph Convolution

- In **spatial graph convolution** used information's from the neighbors of the nodes.

- It don’t have commons with the spectral convolution, therefore it have not the ability to take information's in the **frequency domain**.

Graph Neural Networks

- Introduction to Graphs
- Neural Networks
- Graph Convolutional Networks (GCN)
- **Recurrent Graph Neural Networks (RGNN)**
- Graph Auto-Encoders
- Spatial-Temporal Graph Neural Networks
- GNN Applications

RGNN

- For problems that the data are **sequential** $x(n): \mathbb{Z} \rightarrow S$ we are using **Recurrent Graph Neural Networks.**

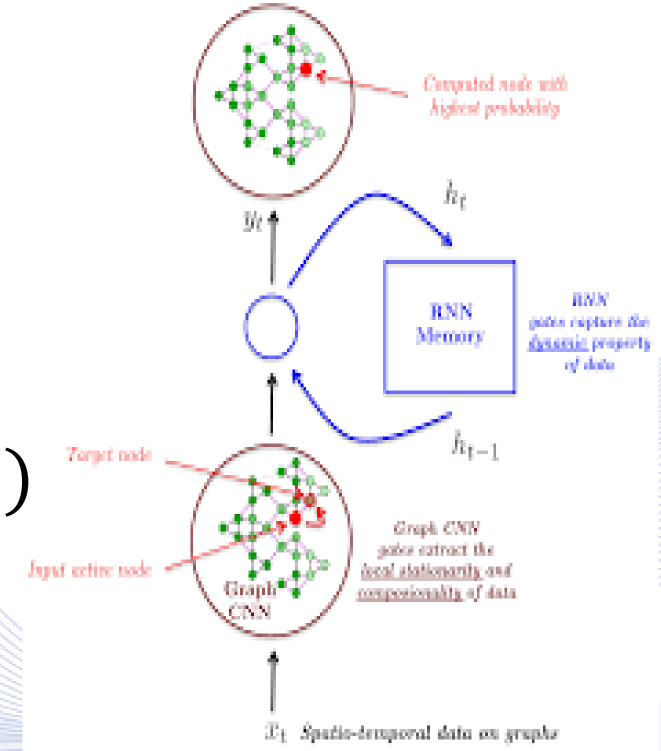
- The **Recurrent Graph Neural Networks** is a combination of the **recurrent neural networks** and the **graph signal processing**.

Graph Convolutional Recurrent Network

- There is and another type of **graph recurrent neural network** the **graph convolutional recurrent network** (**GCRN**).
- While Graph LSTM applies matrix multiplications to its data, graph convolutional recurrent network use graph convolution both to spatial and temporal data.

Graph Convolutional Recurrent Network

- $f = \sigma\left(W_{xf} * x_t + W_{hf} * h_{t-1} + w_{cf} \odot c_{t-1} + \ b_f\right)$
- $\mathrm{o} \ = \ \sigma(W_{xo} * x_t + W_{ho} * h_{t-1} + w_{co} \odot c_{t-1} + \ b_o)$
- $i \ = \ \sigma(W_{xi} * x_t + W_{hi} * h_{t-1} + w_{ci} \odot c_{t-1} + \ b_i)$
- $c_t = f_t \odot c_{t-1} + i_t \odot \tanh(W_{xc} * x_t + W_{hc} h_{t-1} + b_c)$



[3] Youngjoo Seo, Michael Defferrard, Pierre Vandergfheynst, Xavier Bresson, Structured Sequence Modeling with Graph Convolutional Recurrent Networks, 2017

Graph Neural Networks

- Introduction to Graphs
- Neural Networks
- Graph Convolutional Networks (GCN)
- Recurrent Graph Neural Networks (RGNN)
- **Graph Auto-Encoders**
- Spatial-Temporal Graph Neural Networks
- GNN Applications

Graph Auto-Encoders

- The **Graph Auto-Encoders** (**GAE**) are used for **clustering**, **link prediction**, **matrix completion** and **recommendation**.

- They transform the nodes of the input graph into a **vector space** and after reconstruct this information.

Graph Auto-Encoders

- The target of the encoder, by passing the input graph of the **convolutional layers**, is to change each node into an **embedding**.
- Then, the representation of the nodes of the graph can be learned more easily.
- The training of the model captures the information of the topology of each node.

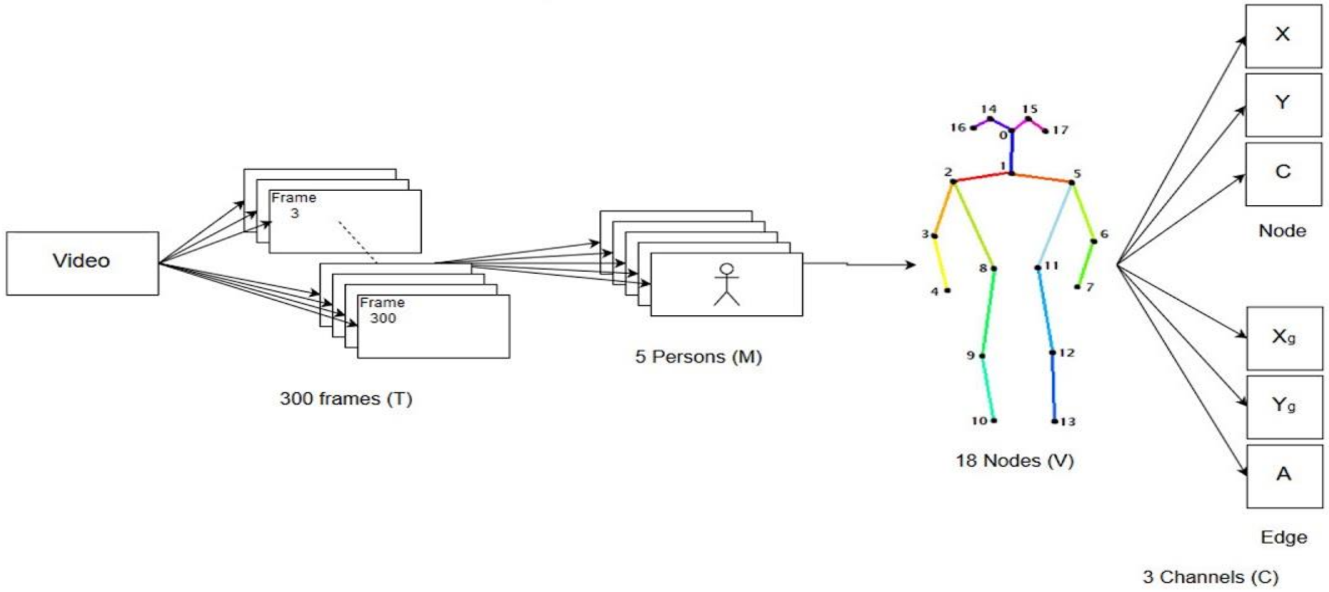
Graph Neural Networks

- Introduction to Graphs
- Neural Networks
- Graph Convolutional Networks (GCN)
- Recurrent Graph Neural Networks (RGNN)
- Graph Auto-Encoders
- **Spatial-Temporal Graph Neural Networks**
- GNN Applications

Spatial-Temporal Graph Neural Networks

- This type of graph neural network make predictions for both **spatial** and **temporal** data(e.g. video).
- The nodes in the **hidden layers** represented by the their **neighbor** nodes.

ST-GCN : Model inputs



Graph Neural Networks

- Introduction to Graphs
- Neural Networks
- Graph Convolutional Networks (GCN)
- Recurrent Graph Neural Networks (RGNN)
- Graph Auto-Encoders
- Spatial-Temporal Graph Neural Networks
- **GNN Applications**

GNN Applications

- Graph Neural Networks have a lot of applications in **life sciences**, the **tech** world and our daily normal life generally.

- In the next slides I will present only few of them like **Chemistry and Biology**, **Social Networks**, **Text Classification**.

GNN Applications

Chemistry and Biology

- **Graph Neural Networks** have an important impact in drug design.
- By computing **molecular fingerprints**(**feature vectors**), which is the representation of the moleculars.
- The fingerprints can be created by an **one-hot vector**, which its digits define the presence or not of a particular substructure.

GNN Applications

Chemistry and Biology [4]

- The **proteins** can be represented with graphs. The **nodes** will be the **amino acid** and for **edges** the **interface** between them.
- Using graph convolution and relation network can be detected breast cancer subtype in classification problem from protein structure.

GNN Applications

Recommendation systems

- The basic architecture of a social recommendation system consist of three parts:
  - **User modeling:** graph modeling in order to learn the latent factor of the ***users***.
  - **Item modeling:** graph modeling in order to learn the latent factor of the ***items***.
  - **Rating prediction:** the task of this part is to find the ideal parameters of the two previous models.

GNN Applications

Text Classification

- ***Text Classification*** is one of the most important problem in the area of Natural Language Processing (NLP)***.***
- Each word in a text ***convert to node*** using **Graph Convolutional Neural Networks** and then with **LSTM** achieve to encode the meaning of the text. [6]
- Similar to **text classification** approaches could also do and **semantic classification**.

Bibliography

[PIT2016] I. Pitas (editor), “Graph analysis in social media”, CRC Press, 2016.
[PIT2021] I. Pitas, “Computer vision”, Createspace/Amazon, in press.
[PIT2017] I. Pitas, “Digital video processing and analysis” , China Machine Press, 2017 (in Chinese).
[PIT2013] I. Pitas, “Digital Video and Television” , Createspace/Amazon, 2013.
[NIK2000] N. Nikolaidis and I. Pitas, “3D Image Processing Algorithms”, J. Wiley, 2000.
[PIT2000] I. Pitas, “Digital Image Processing Algorithms and Applications”, J. Wiley, 2000.

Q & A

Thank you very much for your attention!

More material in
http://icarus.csd.auth.gr/cvml-web-lecture-series/

Contact: Prof. I. Pitas
pitas@csd.auth.gr