Self Study

Based on my Understanding of the Subject

# Graph Neural Network

Compilation of my study materials

**Author:** Dr. Md Arafat Hossain Khan

**Last updated :** November 30, 2022

# Contents

Chapter 1

# Fundamentals

> *Artificial intelligence is a tool, not a threat.Artificial intelligence is a tool, not a threat.Artificial intelligence is a tool, not a threat.*
>
> *Rodney Brooks*

## 1.1 Graph Neural Network Model Mathematics

This note is fundametally inspired by the original paper of graph neural network [1] and based on my undertsanding of the subject.

### 1.1.1 Labels of a graph

A graph $G$ is a pair $(N, E)$, where $N = \{n_1, n_2, \dots, n_{|N|-1}, n_{|N|}\}$ is the set of nodes and $E = \{e_1, e_2, \dots, e_{|E|-1}, e_{|E|}\}$ is the set of edges. $\mathcal{N}(n)$ stands for the neighbors of $n \in N$. If an *undirected* edge $e$ connects node $x$ and $y$, we can also denote $e$ as $(x, y)$ or $(y, x)$. All nodes are embedded into some euclidean space of dimention $d_N \in \mathbb{N}$ and all edges are embedded into some euclidean space of dimention $d_E \in \mathbb{N}$, e.g.

$$\text{label (embedding) for node } n \text{ is defined as } \ \ell_n : n \mapsto \mathbb{R}^{d_N}, \quad \text{where } n \in N$$

$$\text{label (embedding) for edge } e \text{ is defined as } \ \ell_e : e \mapsto \mathbb{R}^{d_E}, \quad \text{where } e \in E$$

The notion of labels can be extended to multiple nodes and edges. Suppose, we take a node set, $\mathfrak{N} = \{\mathfrak{n}_1, \mathfrak{n}_2, \dots, \mathfrak{n}_{|\mathfrak{N}|-1}, \mathfrak{n}_{|\mathfrak{N}|}\} \subseteq N$. Then we define,

$$\ell_{\mathfrak{N}} : \mathbb{R}^{d_N \times |\mathfrak{N}|} \mapsto \mathbb{R}^{d_N} \ \text{ as } \left(\ell_{\mathfrak{n}_1}, \ell_{\mathfrak{n}_2}, \dots, \ell_{\mathfrak{n}_{|\mathfrak{N}|-1}}, \ell_{\mathfrak{n}_{|\mathfrak{N}|}}\right) \mapsto \mathbb{R}^{d_N}$$

Here, we want $\ell_{\mathfrak{N}}$ to be permutation invariant. Similarly, we take an edge set, $\mathfrak{E} = \{\mathfrak{e}_1, \mathfrak{e}_2, \ldots, \mathfrak{e}_{|\mathfrak{E}|-1}, \mathfrak{e}_{|\mathfrak{E}|}\} \subseteq E$. Then we define,

$$\ell_{\mathfrak{E}} : \mathbb{R}^{d_E \times |\mathfrak{E}|} \mapsto \mathbb{R}^{d_E} \quad \text{as} \quad \left(\ell_{\mathfrak{e}_1}, \ell_{\mathfrak{e}_2}, \ldots, \ell_{\mathfrak{e}_{|\mathfrak{E}|-1}}, \ell_{\mathfrak{e}_{|\mathfrak{E}|}}\right) \mapsto \mathbb{R}^{d_E}$$

Here, we want $\ell_{\mathfrak{E}}$ to be permutation invariant. We extend the notion of the labels even further for the entire graph by the following permutation invariant map,

$$\ell : \mathbb{R}^{d_N \times |N|} \times \mathbb{R}^{d_E \times |E|} \mapsto \mathbb{R}^D \quad \text{as} \quad \left(\ell_{n_1}, \ell_{n_2}, \ldots, \ell_{|N|-1}, \ell_{|N|}, \ell_{e_1}, \ell_{e_2}, \ldots, \ell_{|E|-1}, \ell_{|E|}\right) \mapsto \mathbb{R}^D$$

where $D \in \mathbb{N}$.

### 1.1.2 Local transition and output functions

Let us take $n \in N$ from the graph $G = (N, E)$ and define the following,

- $\ell_n \in \mathbb{R}^{d_N}$: node label of $n \in N$.
- $\ell_{\mathcal{N}(n)} \in \mathbb{R}^{d_N}$: label of the neighborhood $\mathcal{N}(n) = \{\nu_1, \nu_2, \ldots, \nu_{|\mathcal{N}(n)|-1}, \nu_{|\mathcal{N}(n)|}\} \subseteq N$.
- $\mathcal{E}(n)$: edges connected to $n$ i.e $\mathcal{E}(n) = \{(\nu_1, n), (\nu_2, n), \ldots, (\nu_{|\mathcal{N}(n)|-1}, n), (\nu_{|\mathcal{N}(n)|}, n)\}$.
- $\ell_{\mathcal{E}(n)}$: edge label of $\mathcal{E}(n)$.

Now we can define something called the *state* of $n$, denoted as $x_n \in \mathbb{R}^s$, which captures the the local information of $n$ using its own node label $\ell_n \in \mathbb{R}^{d_N}$, node label of the neighbors $\ell_{\mathcal{N}(n)} \in \mathbb{R}^{d_N}$, connected edges lalel $\ell_{\mathcal{E}(n)}$, and the *state* of the neighbors $x_{\mathcal{N}(n)} \in \mathbb{R}^s$. Note that, $x_{\mathcal{N}(n)}$ is a permutation invariant map, defined by

$$x_{\mathcal{N}(n)} : \mathbb{R}^{s \times |\mathcal{N}(n)|} \mapsto \mathbb{R}^s \quad \text{as} \quad \left(x_{\nu_1}, x_{\nu_2}, \ldots, x_{\nu_{|\mathcal{N}(n)|-1}}, x_{\nu_{|\mathcal{N}(n)|}}\right) \mapsto \mathbb{R}^s.$$

Then we define the the *local transition function*, that expresses the dependence of a node on its neighborhood as

$$\begin{aligned} f_w : \mathbb{R}^{d_N} \times \mathbb{R}^{d_E} \times \mathbb{R}^s \times \mathbb{R}^{d_N} \mapsto \mathbb{R}^s \quad \text{as} \quad \left(\ell_n, \ell_{\mathcal{E}(n)}, x_{\mathcal{N}(n)}, \ell_{\mathcal{N}(n)}\right) \mapsto \mathbb{R}^s. \\ \text{where} \quad \left(\ell_{(\nu_1,n)}, \ell_{(\nu_2,n)}, \ldots, \ell_{(\nu_{|\mathcal{N}(n)|-1},n)}, \ell_{(\nu_{|\mathcal{N}(n)|},n)}\right) := \ell_{\mathcal{E}(n)} \\ \left(x_{\nu_1}, x_{\nu_2}, \ldots, x_{\nu_{|\mathcal{N}(n)|-1}}, x_{\nu_{|\mathcal{N}(n)|}}\right) := x_{\mathcal{N}(n)} \\ \left(\ell_{\nu_1}, \ell_{\nu_2}, \ldots, \ell_{\nu_{|\mathcal{N}(n)|-1}}, \ell_{\nu_{|\mathcal{N}(n)|}}\right) := \ell_{\mathcal{N}(n)} \end{aligned}$$

And we also express *the local output function* that describes how the output is produced as follows,

$$g_w : \mathbb{R}^s \times \mathbb{R}^{d_N} \mapsto \mathbb{R}^r \quad \text{as} \quad (x_n, \ell_n) \mapsto \mathbb{R}^r$$

Hence, we have the following model,

$$\begin{aligned} \textit{local transition function}, \quad x_n &= f_w\left(\ell_n, \ell_{\mathcal{E}(n)}, x_{\mathcal{N}(n)}, \ell_{\mathcal{N}(n)}\right) \\ \textit{local output function}, \quad o_n &= g_w(x_n, \ell_n) \end{aligned}$$

### 1.1.3 Model Variation

### Minimal model

**Definition 1: Minimal Model**

A model is said to be minimal if it has the smallest number of variables while retaining the same computational power.

Note that, one may wish to remove the labels $\ell_{\mathcal{N}(n)}$, since they include information that is implicitly contained in $x_{\mathcal{N}(n)}$. Thus one can hunt for a minimal model. It can be shown that the minimal model exists but not unique.

### Extension of neighborhood

The neighborhood could contain nodes that are two or more links away from $n$ capturing a bigger local region.

### Directed and mixed graph

For directed or mixed graph we may find a variation of model definition by incorporating the following map into the model definition,

$$\mathfrak{d}_{\mathcal{E}(n)} : \mathcal{E}(n) \mapsto \{0,1\} \quad \text{as}$$

$$\mathfrak{d}_{\mathcal{E}(n)}(e) = \mathfrak{d}_e = \begin{cases} 1 & \text{if } e \in \mathcal{E}(n) \text{ directs towards } n \\ 0 & \text{if } e \in \mathcal{E}(n) \text{ comes out of } n \end{cases}$$

Thus the definition of *local transition function* changes as follows,

$$f_w : \mathbb{R}^{d_N} \times \mathbb{R}^{d_E} \times \mathbb{R}^s \times \mathbb{R}^{d_N} \times \{0,1\}^{|\mathcal{N}(n)|} \mapsto \mathbb{R}^s \quad \text{as} \quad \left(\ell_n, \ell_{\mathcal{E}(n)}, x_{\mathcal{N}(n)}, \ell_{\mathcal{N}(n)}, \mathfrak{d}_{\mathcal{E}(n)}\right) \mapsto \mathbb{R}^s.$$

where

$$\left(\ell_{(\nu_1,n)}, \ell_{(\nu_2,n)}, \dots, \ell_{(\nu_{|\mathcal{N}(n)|-1},n)}, \ell_{(\nu_{|\mathcal{N}(n)|},n)}\right) := \ell_{\mathcal{E}(n)}$$

$$\left(x_{\nu_1}, x_{\nu_2}, \dots, x_{\nu_{|\mathcal{N}(n)|-1}}, x_{\nu_{|\mathcal{N}(n)|}}\right) := x_{\mathcal{N}(n)}$$

$$\left(\ell_{\nu_1}, \ell_{\nu_2}, \dots, \ell_{\nu_{|\mathcal{N}(n)|-1}}, \ell_{\nu_{|\mathcal{N}(n)|}}\right) := \ell_{\mathcal{N}(n)}$$

$$\left(\mathfrak{d}_{(\nu_1,n)}, \mathfrak{d}_{(\nu_2,n)}, \dots, \mathfrak{d}_{(\nu_{|\mathcal{N}(n)|-1},n)}, \mathfrak{d}_{(\nu_{|\mathcal{N}(n)|},n)}\right) := \mathfrak{d}_{\mathcal{E}(n)}$$

However, unless explicitly stated, all the results proposed in this paper hold also for directed graphs and for graphs with mixed directed and undirected links.

### Node specific local transition and output functions

Moreover, each node can have its own *local transition function* and *the local output function* definition,

$$\textit{local transition function,} \quad x_n = f^{(n)}_{w_n}\left(\ell_n, \ell_{\mathcal{E}(n)}, x_{\mathcal{N}(n)}, \ell_{\mathcal{N}(n)}\right)$$

$$\textit{local output function,} \quad o_n = g^{(n)}_{w_n}(x_n, \ell_n)$$

However, for the sake of simplicity, our analysis will consider a particular model where all the nodes share the same implementation.

## Positional and nonpositional graphs

Nonpositional graphs are those described so far. Positional graphs differ since a unique integer identifier is assigned to each neighbors of a node to indicate its logical position.

**Definition 2: Positional graphs**

If for a graph $G = (N, E)$ and for each $n \in N$, there exists,

$$\rho_n : \mathcal{N}(n) \hookrightarrow [|N|]$$

then $G$ is called a positional graph. Note that the symbol $\hookrightarrow$ means an injection and $[|N|]$ means the set $\{1, 2, \dots, |N|\}$.

For $n \in N$,

$$\rho_n(u) = i \implies u \text{ is the } i\text{-th neighbor of } n.$$

An example of this assignment can be such that $\rho_n$ might enumerate the neighbors of a node following a clockwise ordering convention. Notice that for nodes $n_j$ and $n_k$ we may have $\rho_{n_j} = \rho_{n_k}$ which implies that this is only a relative positional assignment. Note that, for a complete graph, $\rho_n$ is a bijection for all $n$, on the other hand for a path graph with at least 3 nodes $\rho_n$ is an injection not a surjection for any $n$.

Now, $f_w$ will take this positional information. Here is how it is done. Suppose, for a graph $G = (N, E)$,

$$M := \max_{n,u} \rho_n(u)$$

Then $\forall n \in N$, we make $x_{\mathcal{N}(n)} = (y_1, y_2, \dots, y_M)$ where,

$$y_i = \begin{cases} x_u & \text{if } \rho_n(u) = i \implies u \text{ is the } i\text{-th neighbor of } n \\ x_0 & \text{if there is no } i\text{-th neighbor of } n \end{cases}$$

Here $x_0$ is some predefined null state. In the same way $\ell_{\mathcal{E}(n)}$ and $\ell_{\mathcal{N}(n)}$ are modified as well.

## Positional and non-positional form

Note that in general,

$$\textit{local transition function}, \quad x_n = f_w\left(\ell_n, \ell_{\mathcal{E}(n)}, x_{\mathcal{N}(n)}, \ell_{\mathcal{N}(n)}\right)$$

For non-positional graph we assume a significant Simplification of the mdoel which turned out to be useful,

$$\textit{local transition function}, \quad x_n = \sum_{\nu_i \in \mathcal{N}(n)} h_w\left(\ell_n, \ell_{(\nu_i, n)}, x_{\nu_i}, \ell_{\nu_i}\right)$$

These two representations are called *positional form* and *non-positional form* respectively.

### 1.1.4 Global transition and output functions

Note that we have *local transition functions* for each node of the graph $G = (N, E)$, where $N = \{n_1, n_2, \ldots, n_{|N|-1}, n_{|N|}\}$ is the set of nodes and $E = \{e_1, e_2, \ldots, e_{|E|-1}, e_{|E|}\}$ is the set of edges.

$$\begin{aligned} x_{n_1} &= f_w\left(\ell_{n_1}, \ell_{\mathcal{E}(n_1)}, x_{\mathcal{N}(n_1)}, \ell_{\mathcal{N}(n_1)}\right) \\ x_{n_2} &= f_w\left(\ell_{n_2}, \ell_{\mathcal{E}(n_2)}, x_{\mathcal{N}(n_2)}, \ell_{\mathcal{N}(n_2)}\right) \\ &\vdots \\ x_{n_{|N|}} &= f_w\left(\ell_{n_{|N|}}, \ell_{\mathcal{E}(n_{|N|})}, x_{\mathcal{N}(n_{|N|})}, \ell_{\mathcal{N}(n_{|N|})}\right) \end{aligned}$$

We define a *global transition function*, $F_w$ that takes the graph $G$ as input and returns the state $x_n$ for each $n \in N$.

Now, observe the *local output functions*,

$$\begin{aligned} o_{n_1} &= g_w(x_{n_1}, \ell_{n_1}) \\ o_{n_2} &= g_w(x_{n_2}, \ell_{n_2}) \\ &\vdots \\ o_{n_{|N|}} &= g_w(x_{n_{|N|}}, \ell_{n_{|N|}}) \end{aligned}$$

We define a *global output function*, $G_w$ that takes the graph $G$ as input and returns the output $o_n$ for each $n \in N$.

At this point we have to make sure that such *global transition function* and *global output function* exists and are unique. Moreover, we also need to figure out a method of computation for the solution.

### Existence and uniqueness of global transition and output functions

To undertsand the existence, uniqueness and calculation method, we first need to undertsand *contraction mapping theorem*, Appendix A.

### 1.1.5 Learning set

Let us sonsider an arbitrary set of graphs $\mathcal{G}$. We take a finite set of graphs from $\mathcal{G}$ as,

$$\{G_1, G_2, \ldots, G_i, \ldots, G_{p-1}, G_p\} \subseteq \mathcal{G} \implies p \leq |\mathcal{G}|.$$

Let us take a subset of the node set of $G_i = (N_i, E_i)$ as,

$$\mathsf{N} = \{n_{i,1}, n_{i,2}, \ldots, n_{i,j}, \ldots, n_{i,q_i-1}, n_{i,q}\} \subseteq N_i \implies q \leq |N_i|.$$

**Definition 3: Domain of a supervised learning framework**

The domain is the set of pairs of a graph and a node, i.e. $\mathcal{D} = \mathcal{G} \times \mathsf{N}$.

We consider a desired target association $t_{i,j} \in \mathbb{R}^m$ of the node $n_{i,j}$, which is a vector in some euclidean space of dimension $m$.

**Definition 4: Learning Set**

We assume a supervised learning framework with the following set,

$$\mathcal{L} = \{(G_i, n_{i,j}, t_{i,j}) | G_i = (N_i, E_i) \in \mathcal{G}, n_{i,j} \in N_i, t_{i,j} \in \mathbb{R}^m, 1 \leq i \leq p, 1 \leq j \leq q_i\}$$

$\mathcal{L}$ is called the learning set.

Note that $\mathcal{G}$ or $\{G_1, G_2, \ldots, G_i, \ldots, G_{p-1}, G_p\}$ can also be considered as one single graph which may have multiple connected components and modify the definition of learning set accordingly.

Appendix A

# Contraction mapping theorem

**Definition 5: Contraction map**

Let $(X, d)$ be a complete metric space. Then a map $T : X \to X$ is called a contraction map on $X$ if there exists $q \in [0, 1)$ such that

$$d\,(T(x), T(x')) \leq q\, d(x, x')$$

for all $x, x' \in X$. Contraction map is also known as contractive map. If the above condition is instead satisfied for $q \in [0, 1]$, then the mapping is said to be a non-expansive map.
More generally, if $(X, d_X)$ and $(Y, d_Y)$ are two metric spaces, then $T : X \to Y$ is a contractive mapping if there is a constant $q \in [0, 1)$ such that

$$d_Y(T(x), T(x')) \leq q\, d_X(x, x')$$

for all $x$ and $x'$ in $X$.

**Definition 6: Lipschitz continuous map**

Given two metric spaces $(X, d_X)$ and $(Y, d_Y)$, a function $f : X \to Y$ is called Lipschitz continuous if there exists a real constant $K \geq 0$ such that, for all $x$ and $x'$ in $X$,

$$d_Y(f(x), f(x')) \leq K d_X(x, x').$$

Any such $K$ is referred to as a Lipschitz constant for the function $f$ and $f$ may also be referred to as $K$-Lipschitz.

- The smallest constant is sometimes called the (best) Lipschitz constant of $f$ or the dilation or dilatation of $f$.
- If $K = 1$ the function is called a short map.
- If $0 \leq K < 1$, the function is called a contraction [Definition 5].

**Lemma A.1**

Contraction map $\implies$ Lipschitz continuous map.

*Proof.* Clearly by definition. ■

### Definition 7: Uniformly continuous map

Let $M_1 = (A_1, d_1)$ and $M_2 = (A_2, d_2)$ be metric spaces. Then a mapping $f : A_1 \to A_2$ is uniformly continuous on $A_1$ if and only if:

$$\forall \epsilon \in \mathbb{R}_{>0} : \exists \delta \in \mathbb{R}_{>0} : \forall x, y \in A_1 : d_1(x, y) < \delta \implies d_2(f(x), f(y)) < \epsilon.$$

### Definition 8: Continuous map

Let $M_1 = (A_1, d_1)$ and $M_2 = (A_2, d_2)$ be metric spaces. Then a mapping $f : A_1 \to A_2$ is continuous at a point $c \in A_1$ if and only if:

$$\forall \epsilon \in \mathbb{R}_{>0} : \exists \delta \in \mathbb{R}_{>0} : d_1(c, y) < \delta \implies d_2(f(c), f(y)) < \epsilon.$$

### Lemma A.2

Uniformly continuous map $\implies$ Continuous map.

*Proof.* Given $\epsilon > 0$. Then by uniform continuity, there is $\delta$ such that for all $x, y \in A_1$ we have

$$d_1(x, y) < \delta \implies d_2(f(x), f(y)) < \epsilon.$$

In particular if we take $x = c$ we get that for all $y \in A_1$

$$d_1(x, y) < \delta \implies d_2(f(x), f(y)) < \epsilon.$$

This is the definition of continuity [Definition 8]. ■

### Lemma A.3

Lipschitz continuous map $\implies$ Uniformly continuous map.

*Proof.* Given two metric spaces $(X, d_X)$ and $(Y, d_Y)$, a Lipschitz continuous map $f : X \to Y$, we get from Definition 6, there exists a real constant $K \geq 0$ such that, for all $x$ and $x'$ in $X$,

$$d_Y(f(x), f(x')) \leq K d_X(x, x').$$

- Case 1 ($K = 0$): we get for all $x$ and $x'$ in $X$,

  $$d_Y(f(x), f(x')) = 0.$$

  Hence, $f$ is a constant map. Clearly $f$ is uniformly continuous [use Definition 6].

- Case 2 ($K > 0$): By Lipschitz continuity,

  $$\forall x, x' \in X : d_Y(f(x), f(x')) \leq K d_X(x, x').$$

  Take $\epsilon > 0$. Then there exists $\delta = \frac{\epsilon}{K}$. Suppose, $d_X(x, x') < \delta$. Hence, we have,

  $$\begin{aligned} \forall \epsilon \in \mathbb{R}_{>0} : \exists \delta = \frac{\epsilon}{K} \in \mathbb{R}_{>0} : \forall x, x' \in X : & d_X(x, x') < \delta \\ & \implies K d_X(x, x') < \epsilon \\ & \implies d_Y(f(x), f(x')) < \epsilon. \end{aligned}$$

■

### Lemma A.4

Contraction map $\implies$ Lipschitz continuous map $\implies$ Uniformly continuous map $\implies$ Continuous map.

*Proof.* Using Lemma A.1, A.3 and A.2. ■

### Theorem A.1: Contraction mapping theorem

Let $(X, d)$ be a non-empty complete metric space with a contraction mapping $T : X \to X$. Then $T$ admits a unique fixed-point $x^* \in X$ (i.e. $T(x^*) = x^*$). Furthermore, $x^*$ can be found by starting with an arbitrary element $x_0 \in X$ and defining a sequence $(x_n)_{n\in\mathbb{N}}$ by $x_n = T(x_{n-1})$ for $n \geq 1$. Then $\lim_{n\to\infty} x_n = x^*$.

# Index

# Bibliography

[1] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. “The Graph Neural Network Model”. In: *IEEE Transactions on Neural Networks* 20.1 (2009), pp. 61–80. DOI: 10.1109/TNN.2008.2005605 (page - 1).