

## 5 - Νευρωνικά δίκτυα και γραφήματα - γενίκευση

Δημήτριος Κοσμόπουλος

Πανεπιστήμιο Πατρών  
Τμήμα Μηχανικών ΗΥ κ Πληροφορικής

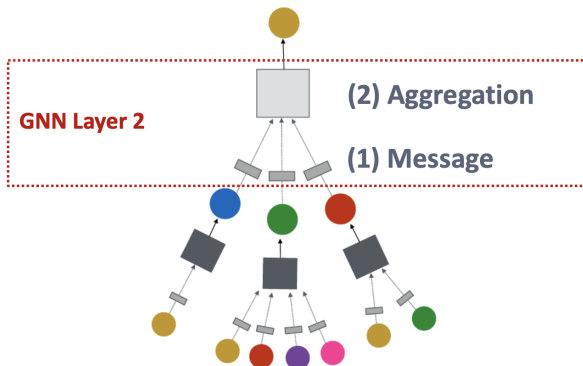
18 Δεκεμβρίου 2023

# Ένα επίπεδο GNN

Το επίπεδο (GNN) αποτελείται από:

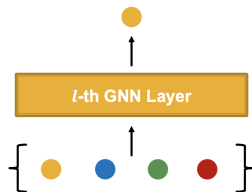
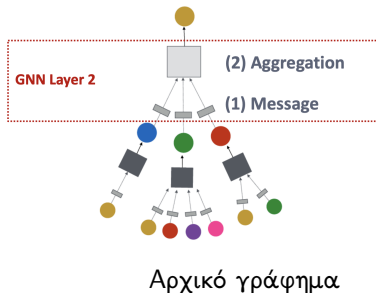
- ▶ μήνυμα
- ▶ συνένωση μηνυμάτων

Οι διαφορετικές υλοποιήσεις (GNN) υλοποιούν διαφορετικά αυτά τα δύο.



# Ένα επίπεδο GNN

Είσοδος από τον ίδιο κόμβο και τους γείτονες.



Είσοδος:  $\mathbf{h}_v^{l-1}, \mathbf{h}_{u \in N(v)}^{l-1}$ ,  
Έξοδος:  $\mathbf{h}_v^{(l)}$

# Υπολογισμός μηνύματος (1)

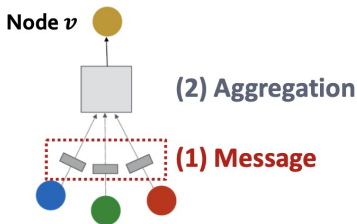
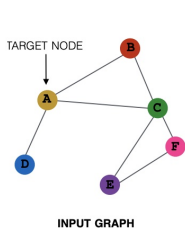
Ιδέα: κάθε κόμβος δημιουργεί ένα μήνυμα που θα σταλεί στους γείτονές του την επόμενη χρονική στιγμή.

Η συνάρτηση είναι της μορφής:

$$\mathbf{m}_u^l = \text{MSG}^l(\mathbf{h}_u^{l-1})$$

Παράδειγμα:

$$\mathbf{m}_u^l = \mathbf{W}^l \mathbf{h}_u^{l-1}$$



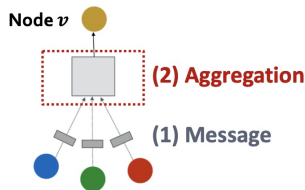
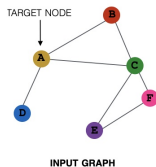
## Συγκερασμός μηνυμάτων (2)

Ιδέα: ο κόμβος  $v$  συνενώνει τα μηνύματα από όλους τους γείτονες  $u$ :

$$\mathbf{h}_v^l = \text{AGG}^l(\mathbf{m}_{u,v}^l, u \in N(v))$$

π.χ.  $\text{sum}()$ ,  $\text{mean}()$ ,  $\text{max}()$

$$\mathbf{h}_v^l = \text{sum}(\mathbf{m}_{u,v}^l, u \in N(v))$$



## Συγκερασμός μηνυμάτων (2)

Πρόβλημα: Η πληροφορία από τον κόμβο  $v$  μπορεί να χαθεί διότι δεν λαμβάνεται υπόψη στον υπολογισμό του  $h'_v$  μέσω συνένωσης το  $h_v'^{-1}$

Λύση: να συμπεριλάβουμε στον υπολογισμό του  $h'_v$  μέσω συνένωσης το  $h_v'^{-1}$

## Συγκερασμός μηνυμάτων (2)

1. Σύνθεση μηνυμάτων Υπολογισμός μηνύματος κόμβου  $v$  από τον εαυτό του. Συνήθως γίνεται διακριτός υπολογισμός σε σχέση με γείτονες



$$\begin{array}{ll} (\alpha') \text{ γείτονες:} & (\beta') \text{ ο κόμβος} \\ \mathbf{m}'_u = \mathbf{W}'\mathbf{h}'_{u-1} & v: \mathbf{m}'_v = \mathbf{B}'\mathbf{h}'_{v-1} \end{array}$$

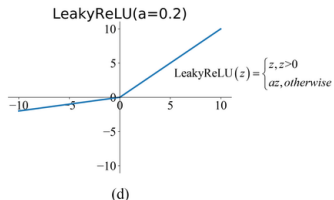
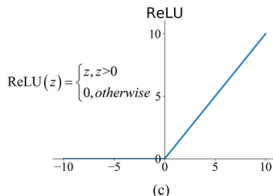
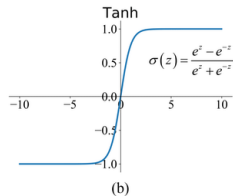
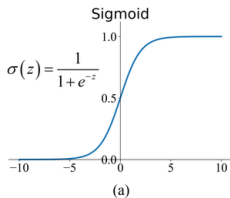
Τα βάρη των γειτόνων είναι κοινά, του κόμβου  $v$  είναι διαφορετικό.

2. Μετά το συγκερασμό μηνυμάτων των γειτόνων μπορούμε να συνενώσουμε και το μήνυμα του  $v$  μέσω άθροισης ή συνένωσης (concatenation), π.χ.:

$$\mathbf{h}'_v = \text{CONCAT}(\text{AGG}(\{\mathbf{m}'_u, u \in N(v)\}), \mathbf{m}'_v)$$

# Μη γραμμικότητες

- ▶ Μπορούν να προστεθούν τόσο στο μήνυμα όσο και στον συγκερασμό
- ▶ Συνήθως αναπαριστώνται με  $\sigma(\cdot)$
- ▶ Τυπικές μη γραμμικότητες: *sigmoid*, *tanh*, *ReLU*, *LeakyReLU*





## Τυπικές αρχιτεκτονικές: graph convolutional network - GCN

Η κατάσταση του κόμβου  $v$  τη χρονική στιγμή  $l$  δίνεται από τη σχέση:

$$\mathbf{h}_v^l = \sigma \left( \sum_{u \in N(v)} \mathbf{W}^l \frac{\mathbf{h}_u^{l-1}}{|N(v)|} \right)$$

1. Μήνυμα από κάθε γειτονικό κόμβο  $u$ :

$$\mathbf{m}_u^l = \mathbf{W}^l \mathbf{h}_u^{l-1}$$

Κανονικοποίηση με το βαθμό του κόμβου  $v$

2. Συγκερασμός: Άθροιση των γειτονικών κόμβων, καθώς και του ίδιου κόμβου  $v$  που δεν εξετάζεται χωριστά

$$\mathbf{h}_v^l = \sigma \left( \frac{1}{|N(v)|} \text{Sum} \left( \left\{ \mathbf{m}_u^l, u \in N(v) \right\} \right) \right)$$

## Τυπικές αρχιτεκτονικές: graphSAGE

Η κατάσταση του κόμβου  $v$  τη χρονική στιγμή  $l$  δίνεται από τη σχέση:

$$\mathbf{h}_v^l = \sigma \left( \mathbf{W}^l \cdot \text{CONCAT} \left( \mathbf{h}_v^{(l-1)}, \text{AGG} \left( \left\{ \mathbf{h}_u^{(l-1)}, \forall u \in N(v) \right\} \right) \right) \right)$$

1. Μήνυμα από τους γειτονικούς κόμβους  $u$ : είναι το όρισμα του AGG
2. Συγκερασμός γίνεται σε δύο φάσεις:
  - ▶ Συγκερασμός των γειτονικών κόμβων, καθώς και του ίδιου κόμβου  $v$  που δεν εξετάζεται χωριστά

$$\mathbf{h}_{N(v)}^l \leftarrow \text{AGG} \left( \left\{ \mathbf{h}_u^{l-1}, \forall u \in N(v) \right\} \right)$$

- ▶ Συμπερίληψη του κόμβου  $v$

$$\mathbf{h}_v^l \leftarrow \sigma \left( \mathbf{W}^l \cdot \text{CONCAT} \left( \mathbf{h}_v^{l-1}, \mathbf{h}_{N(v)}^l \right) \right)$$

## Τυπικές αρχιτεκτονικές: graphSAGE

Για συγκερασμό των γειτονικών κόμβων υπάρχουν διάφορες επιλογές.

1. Μέση τιμή:

$$AGG = \sum_{u \in N(v)} \mathbf{W}' \frac{\mathbf{h}'_{u^{l-1}}}{|N(v)|}$$

2. Pooling: Μετασχηματισμός των μηνυμάτων των γειτόνων μέσω *MLP* δικτύου και εφαρμοσε *mean(.)* ή *max(.)*

$$AGG = \text{Mean} \left( \left\{ \text{MLP} \left( \mathbf{h}'_{u^{l-1}} \right), \forall u \in N(v) \right\} \right)$$

3. Εφαρμοσε *LSTM* (δίκτυο για μοντελοποίηση ακολουθίας) στους γείτονες μετά από τυχαία μετάθεση  $\pi$ .

$$AGG = \text{LSTM} \left( \left[ \mathbf{h}'_{u^{l-1}}, \forall u \in \pi(N(v)) \right] \right)$$

Προαιρετικά εφαρμόζουμε κανονικοποίηση στα  $\eta'_u$  διαιρώντας με την  $L_2$  νόρμα.

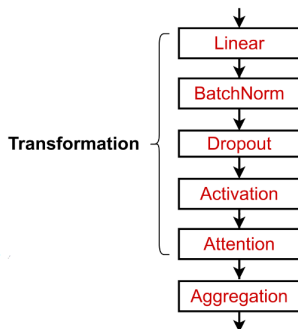
# Τυπικές αρχιτεκτονικές: graphSAGE

$l_2$ -κανονικοποίηση:

- ▶ Προαιρετικά: εφάρμοσε  $l_2$ -κανονικοποίηση σε κάθε επίπεδο  
$$\mathbf{h}'_v \leftarrow \frac{\mathbf{h}_v'}{\|\mathbf{h}_v'\|}$$
- ▶ Χωρίς κανονικοποίηση οι ενσωματώσεις έχουν διαφορετικές κλίμακες
- ▶ Κάποιες φορές οδηγούμαστε σε βελτίωση απόδοσης
- ▶ Μετά την κανονικοποίηση οι ενσωματώσεις έχουν την ίδια  $l_2$  νόρμα

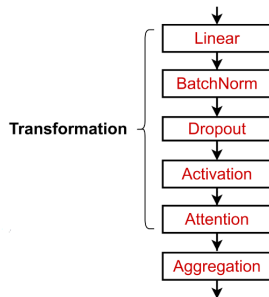
# Θέματα υλοποίησης

- ▶ Τα κλασσικά επίπεδα ενός *GNN* είναι μια πολύ καλή πρώτη προσέγγιση
- ▶ Μπορούμε να πετύχουμε βελτίωση με μια τυπική αρχιτεκτονική
- ▶ Μπορούμε να συμπεριλάβουμε μονάδες που χρησιμοποιούνται στη βαθιά μηχανική μάθηση
- ▶ Η βέλτιστη αρχιτεκτονική είναι πεδίο έρευνας



# Θέματα υλοποίησης

- ▶ Batch normalization: σταθεροποιεί την εκπαίδευση
- ▶ Dropout: προλαμβάνει την υπερεκπαίδευση
- ▶ Attention: ελέγχει το βάρος ενός μηνύματος



## Θέματα υλοποίησης: Batch normalization

- ▶ **Στόχος:** να σταθεροποιεί την εκπαίδευση
- ▶ **Ιδέα:** δεδομένου ενός συνόλου εισόδων (ενσωματώσεων)
  - ▶ Κεντράρισε τις εισόδους ώστε να έχουν μηδενική μέση τιμή
  - ▶ Κανονικοποίησε με κλίμακα ώστε να έχουμε μοναδιαία διασπορά

**Είσοδος:**  $\mathbf{X} \in \mathcal{R}^{N \times D}$ , **Έξοδος:**  $\mathbf{Y}_{i,j} \in \mathcal{R}^{N \times D}$  κανονικοποιημένα διανύσματα

$N$  αριθμός διανυσμάτων ενσωματώσεως,  $D$  η διάσταση

Βήμα 1:  $\mu_j = \frac{1}{N} \sum_{i=1}^N X_{i,j}$      $\sigma_j^2 = \frac{1}{N} \sum_{i=1}^N (X_{i,j} - \mu_j)^2$

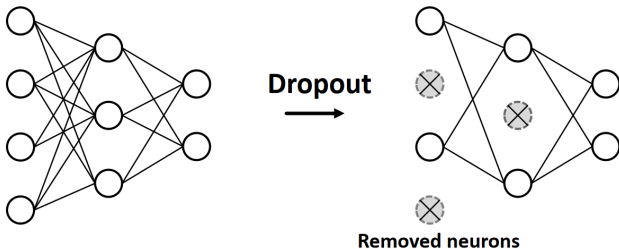
Βήμα 2: Κάνε την κανονικοποίηση

$$\hat{X}_{i,j} = \frac{X_{i,j} - \mu_j}{\sqrt{\sigma_j^2 + \epsilon}} \quad \mathbf{Y}_{i,j} = \gamma_j \hat{X}_{i,j} + \beta_j$$

$\gamma_j, \beta_j$  : παράμετροι προς εκπαίδευση

## Θέματα υλοποίησης: dropout

- ▶ **Στόχος:** κανονικοποίηση ώστε να προλαμβάνει την υπερεκπαίδευση
- ▶ **Ιδέα:**
  - ▶ Κατά την εκπαίδευση μηδένισε κόμβους με πιθανότητα  $p$
  - ▶ Κατά την δοκιμή χρησιμοποίησε όλους τους κόμβους



Για τα  $GNN$  εφαρμόζεται στο γραμμικό επίπεδο της συνάρτησης που φτιάχνει το μήνυμα ( $\mathbf{m}'_u = \mathbf{W}'\mathbf{h}'_{u-1}$ ), δηλαδή μηδενίζοντας τυχαία στήλες στον πίνακα  $\mathbf{W}$ .



## Θέματα υλοποίησης: attention

- ▶ Κάποιοι κόμβοι-γείτονες του  $v$  είναι σημαντικότεροι από άλλους και θα πρέπει να λαμβάνονται υπόψη με μεγαλύτερο βάρος  $\alpha_{vu}$

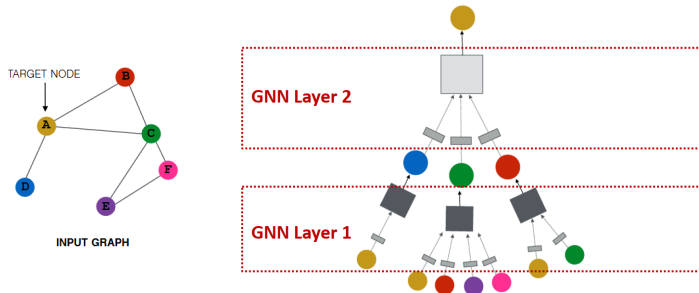
$$\mathbf{h}'_v = \sigma \left( \sum_{u \in N(v)} \alpha_{vu} \mathbf{W}' \mathbf{h}_u^{(l-1)} \right)$$

- ▶ Σε *GCN/GraphSAGE*  $\alpha_{vu} = \frac{1}{|N(v)|}$  δηλαδή το βάρος κάθε γείτονα εξαρτάται
  - ▶ από το βαθμό του  $v$
  - ▶ όλοι οι γείτονες είναι το ίδιο σημαντικοί
- ▶ Μπορούμε να υπολογίσουμε τα βάρη  $\alpha_{vu}$  μέσω εκπαίδευσης.

# Συσσώρευση επιπέδων

Πώς συνδέουμε επίπεδα σε ένα *GNN* ;

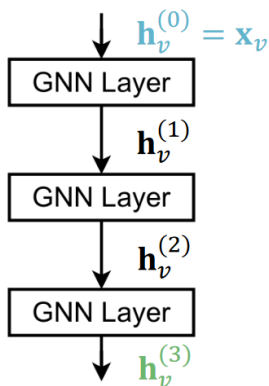
- ▶ σειριακή συσσώρευση επιπέδων
- ▶ συνδέσεις με παράλειψη επιπέδων (skip connections)



## Συσσώρευση επιπέδων

Πώς συνδέουμε επίπεδα σε ένα *GNN* ;

- ▶ τυπικός τρόπος: σειριακή συσσώρευση επιπέδων
- ▶ **είσοδος**: το διανυσμα χαρακτηριστικών του κόμβου  $x_v$
- ▶ **έξοδος**: ενσωμάτωση χαρακτηριστικών  $h_v^l$  έπειτα από  $L$  επίπεδα



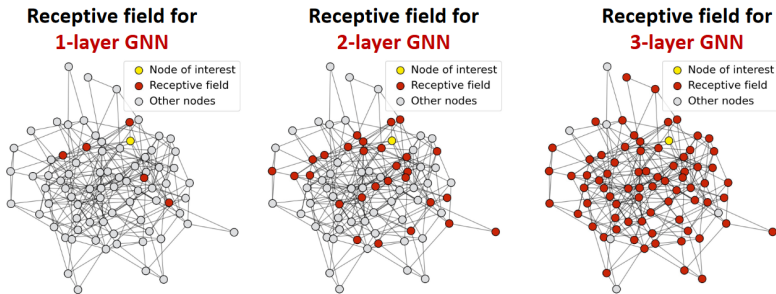
# Το πρόβλημα της εξομάλυνσης

- ▶ Τα *GNN* είναι ευαίσθητα στην υπερ-εξομάλυνση
- ▶ οι έξοδοι συγκλίνουν στην ίδια τιμή ενώ εμείς θέλουμε διαφοροποίηση
- ▶ γιατί συμβαίνει αυτό;

# Δεκτικό πεδίο

- ▶ Δεκτικό πεδίο ενός κόμβου είναι το σύνολο κόμβων που επηρεάζουν την ενσωμάτωση του κόμβου
- ▶ Σε  $GNN$   $k$  επιπέδων κάθε κόμβος έχει δεκτικό πεδίο μια γειτονιά  $k$  βημάτων

π.χ. για ένα κόμβο:

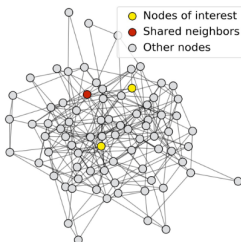


# Δεκτικό πεδίο

Η τομή του δεκτικού πεδίου δύο κόμβων μεγαλώνει πολύ γρήγορα αν αυξήσουμε τον αριθμό των επιπέδων (βημάτων)

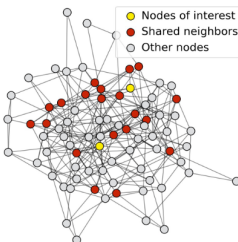
## 1-hop neighbor overlap

Only 1 node



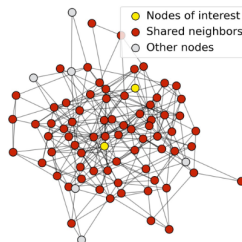
## 2-hop neighbor overlap

About 20 nodes



## 3-hop neighbor overlap

Almost all the nodes!



## Δεκτικό πεδίο και εξομαλυνση

Μπορούμε να εξηγήσουμε την υπερβολική εξομάλυνση μέσω της έννοιας του δεκτικού πεδίου

- ▶ Γνωρίζουμε ότι η ενσωμάτωση ενός κόμβου καθορίζεται από το δεκτικό του πεδίο
- ▶ Εάν δύο κόμβοι έχουν πολύ επικαλυπτόμενα δεκτικά πεδία, τότε οι ενσωματώσεις τους είναι πολύ παρόμοιες
- ▶ Συσσώρευση πολλών επιπέδων  $GNN$   $\rightarrow$  οι κόμβοι θα έχουν πολυεπικαλυπτόμενα δεκτικά πεδία  $\rightarrow$  οι ενσωματώσεις κόμβων θα είναι πολύ παρόμοιες  $\rightarrow$  υποφέρουν από την υπερβολική εξομάλυνση

Πώς θα ξεπεράσουμε το πρόβλημα της υπερβολικής εξομάλυνσης;

## Σχεδίαση των επιπέδων *GNN*

Τι μαθαίνουμε από το πρόβλημα της υπερβολικής εξομάλυνσης;

- ▶ **Μάθημα 1:** Πρέπει να είμαστε προσεκτικοί όταν προσθέτουμε επίπεδα *GNN*
- ▶ Σε αντίθεση με τα νευρωνικά δίκτυα σε άλλους τομείς (*CNN* για εικόνα ταξινόμηση), η προσθήκη περισσότερων επιπέδων *GNN* δεν βοηθά πάντα
  - ▶ Βήμα 1: Αναλύστε το απαραίτητο δεκτικό πεδίο για να λύσετε το πρόβλημα πρόβλημα. π.χ., υπολογίζοντας τη διάμετρο του γραφήματος
  - ▶ Βήμα 2: Ορίστε τον αριθμό των επιπέδων *GNN*  $L$  ώστε το δεκτικό πεδίο να είναι λίγο μεγαλύτερο από όσο το χρειαζόμαστε. Μην ορίσετε το  $L$  να είναι άσκοπα μεγάλο!

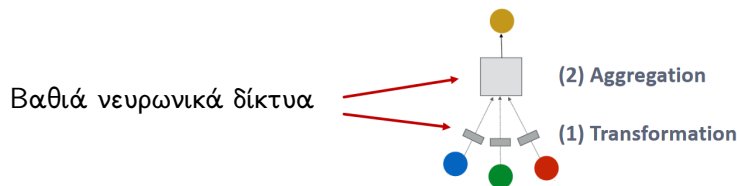
Ερώτηση: Πώς να ενισχύσετε την εκφραστική δύναμη του *GNN*, αν ο αριθμός των επιπέδων *GNN* είναι μικρός;



# Δυνατότητες αναπαράστασης για ρηχά δίκτυα

Πώς να βελτιώσουμε τις δυνατότητες αναπαράστασης για ένα ρηχό *GCN*;

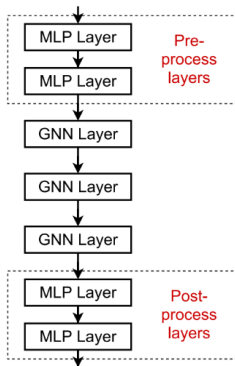
- ▶ **Λύση 1:** Αυξάνουμε την εκφραστική δύναμη μέσα κάθε *GNN* επίπεδο
  - ▶ Στα προηγούμενα παραδείγματα μας, κάθε μετασχηματισμός ή συνάρτηση συγκερασμού περιλαμβάνει μόνο ένα γραμμικό επίπεδο
  - ▶ Μπορούμε να ορίσουμε το μετασχηματισμό ή τη συνάρτηση συγκερασμού ως ένα βαθύ νευρωνικό δίκτυο



# Δυνατότητες αναπαράστασης για ρηχά δίκτυα

## Μάθημα 2: Προσθήκη συνδέσεων παράλειψης σε GCN

- ▶ Επίπεδα προεπεξεργασίας είναι απαραίτητα για μετασχηματισμό χαρακτηριστικών, π.χ., κόμβοι που αναπαριστούν εικόνες/κείμενο
- ▶ Επίπεδα μετα - επεξεργασίας είναι σημαντικά όταν χρειάζονται νέες ενσωματώσεις π.χ. ταξινόμηση γραφημάτων, γραφήματα γνώσης

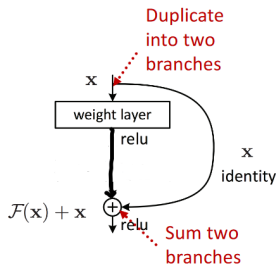
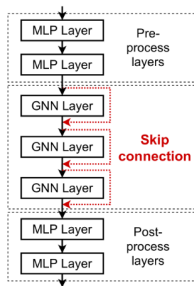


# Σχεδίαση των επιπέδων GNN

Τι γίνεται αν το πρόβλημά εξακολουθεί να απαιτεί πολλά επίπεδα GNN;

**Μάθημα 2:** Προσθήκη συνδέσεων παράλειψης (skip connection)

- ▶ Παρατήρηση από εξομάλυνση: οι ενσωματώσεις κόμβων από τα προηγούμενα επίπεδα GNN μπορούν μερικές φορές να διαφοροποιήσουν καλύτερα τους κόμβους
- ▶ Λύση: Μπορούμε να αυξήσουμε τον αντίκτυπο των προηγούμενων στρωμάτων στις τελικές ενσωματώσεις κόμβων, με την προσθήκη συντομεύσεων
- ▶ Αντί να έχουμε  $\mathcal{F}(x)$  στην έξοδο έχουμε  $\mathcal{F}(x) + x$

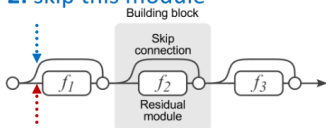


# Τα νευρωνικά δίκτυα στα διάφορα επίπεδα

Γιατί λειτουργούν οι συνδέσεις παράλειψης;

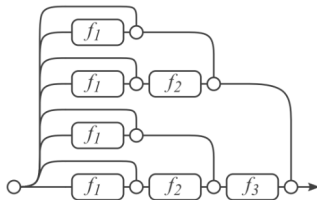
- ▶ Διαίσθηση: η παράλειψη συνδέσεων δημιουργεί ένα μείγμα μοντέλων βαθιών και ρηχών  $GNN$
- ▶  $N$  παραλείψεις συνδέσεων  $\rightarrow 2^N$  πιθανές διαδρομές
- ▶ κάθε μονοπάτι έχει μέχρι  $N$  στοιχεία επεξεργασίας
- ▶ το άθροισμα είναι η μέση τιμή των γειτόνων (αναλλοίωτο ως προς τη μετάθεση)

**Path 2: skip this module**



**Path 1: include this module**

Δίκτυο παράλειψης με 2 μπλοκ



Ανάπτυγμα μονοπατιών

## Δίκτυο παράλειψης

- ▶ Τυπικό επίπεδο :

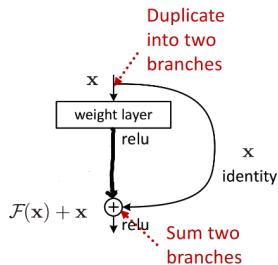
$$\mathbf{h}'_v = \sigma \left( \sum_{u \in N(v)} \frac{1}{|N(v)|} \mathbf{W}' \mathbf{h}'_{u^{l-1}} \right)$$

$F(x)$

- ▶ Με σύνδεση παράλειψης :

$$\mathbf{h}'_v = \sigma \left( \sum_{u \in N(v)} \frac{1}{|N(v)|} \mathbf{W}' \mathbf{h}'_{u^{l-1}} + \mathbf{h}'_{v^{l-1}} \right)$$

$F(x) + x$



# Δίκτυο παράλειψης

- ▶ Άλλες επιλογές: Απευθείας μετάβαση στο τελευταίο επίπεδο
- ▶ Το τελικό στρώμα συγκεράζει απευθείας όλες τα ενσωματώσεις κόμβων στα προηγούμενα επίπεδα

