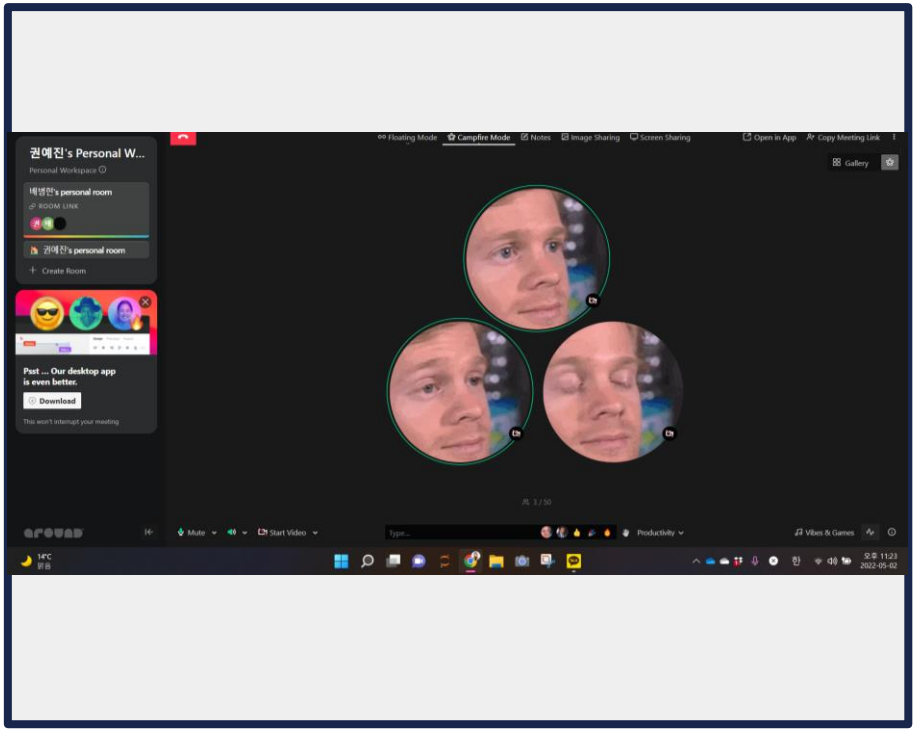
CUAI 스터디 GNN 팀

2022.05.10

발표자 : 권예진

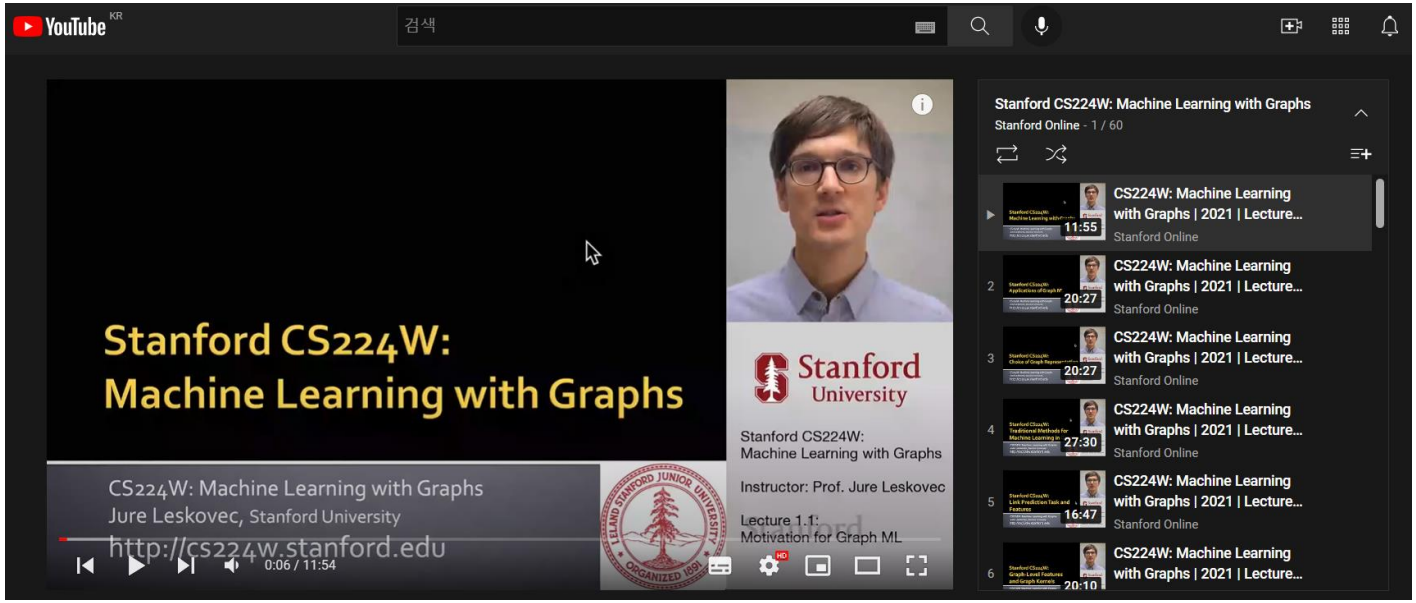
스터디원 소개 및 만남 인증



5.2 10:00 PM
세 번째 스터디

<참석자>
스터디원 : 권예진
스터디원 : 이하윤
스터디원 : 배병현

스터디 소개

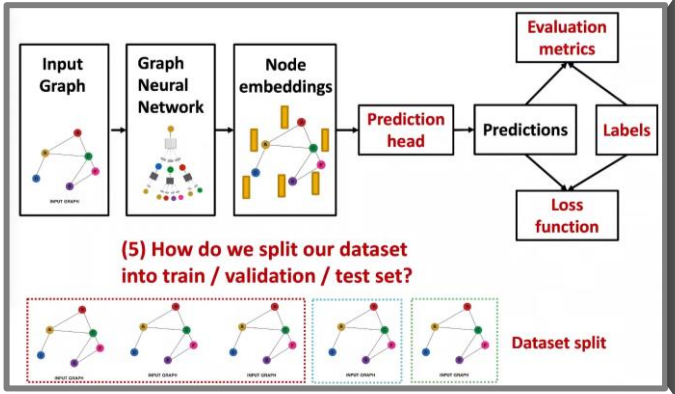


Stanford CS224W: Machine Learning with Graphs

research on the structure and analysis of Graph Neural Network

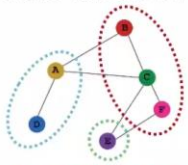
권예진 스터디원 발표

Lecture 8.3 – Setting up GNN Prediction Tasks



Transductive node classification

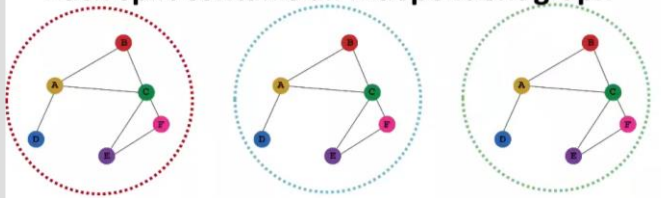
- **All the splits can observe the entire graph structure**, but can only observe the labels of their respective nodes



Training
Validation
Test

Inductive node classification

- Suppose we have a dataset of 3 graphs
- **Each split contains an independent graph**

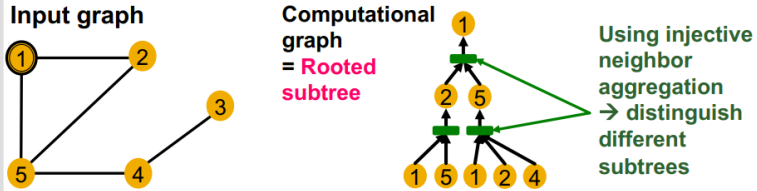


Training
Validation
Test

이하윤 스터디원 발표

Lecture 9.1 – How Expressive are Graph Neural Networks
Lecture 9.2 – Designing the Most Powerful GNNs

To generate a node embedding, GNNs use a computational graph corresponding to a subtree rooted around each node.



GNN can fully distinguish different subtree structures if **every step of its neighbor aggregation is injective.**

Next: We analyze aggregation functions of two popular GNN models

- **GCN** (mean-pool) [Kipf & Welling, ICLR 2017]
  - Uses **element-wise** mean pooling over neighboring node features

$$\text{Mean}(\{x_u\}_{u \in N(v)})$$

aggregation func.

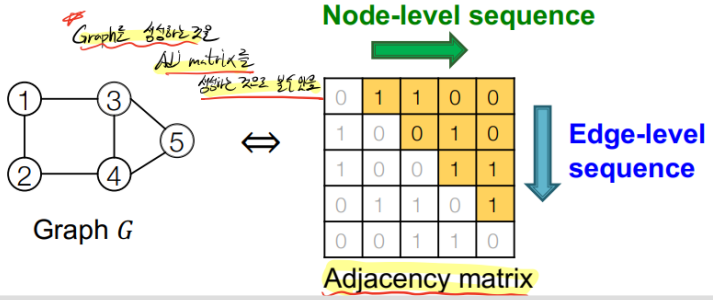
- **GraphSAGE** (max-pool) [Hamilton et al. NeurIPS 2017]
  - Uses **element-wise** max pooling over neighboring node features

$$\text{Max}(\{x_u\}_{u \in N(v)})$$

배병현 스터디원 발표

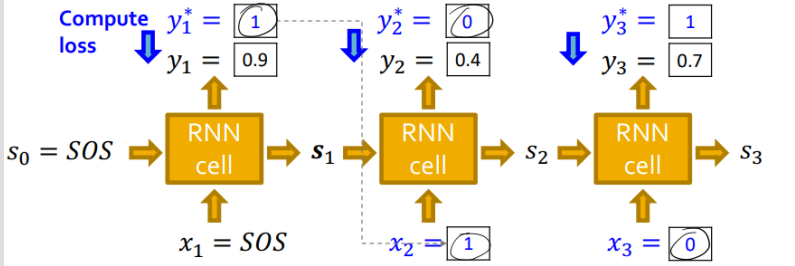
Lecture 15.1 – Deep Generative Models for Graphs
Lecture 15.2 – Graph RNN: Generating Realistic Graphs

- **Summary: A graph + a node ordering = A sequence of sequences**
- Node ordering is randomly selected (we will come back to this)



Training the model:

- We observe a sequence y^* of edges [1,0,...]
- **Principle**: Teacher Forcing -- Replace input and output by the real sequence



감사합니다☺