DialogueGCN: A Graph Convolutional Neural Network for Emotion Recognition in Conversation

Deepanway Chosal, Navonil Majumder, Soujanya Poria, Niyati Chhaya, Alexander Gelbukh

EMNLP 2019

김유리

Index

Introduction

Introduction

Task : Emotion Recognition (Classification)

DialogueGCN 제안

EMNLP 2019

2019년 당시 Emotion Recognition SOTA

Introduction

Emotion Recognition 기존 연구는 전~부 RNN, LSTM 기반

Ex. CMN, ICON, DialogueRNN, …

Introduction

그런데, GCN을 적용 했다?

GCN

GCN

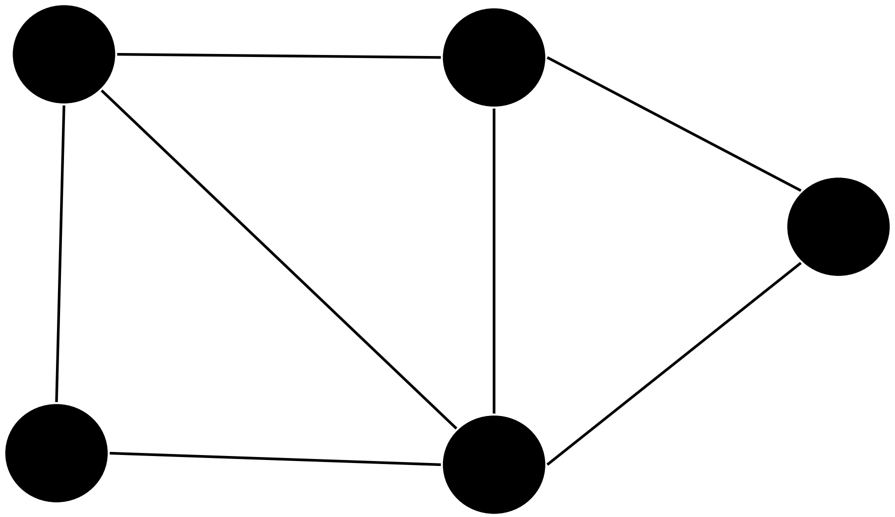
GCN?

Graph **C**onvolution **N**etwork

GCN

Graph structure

Vertex(node), Edge로 구성



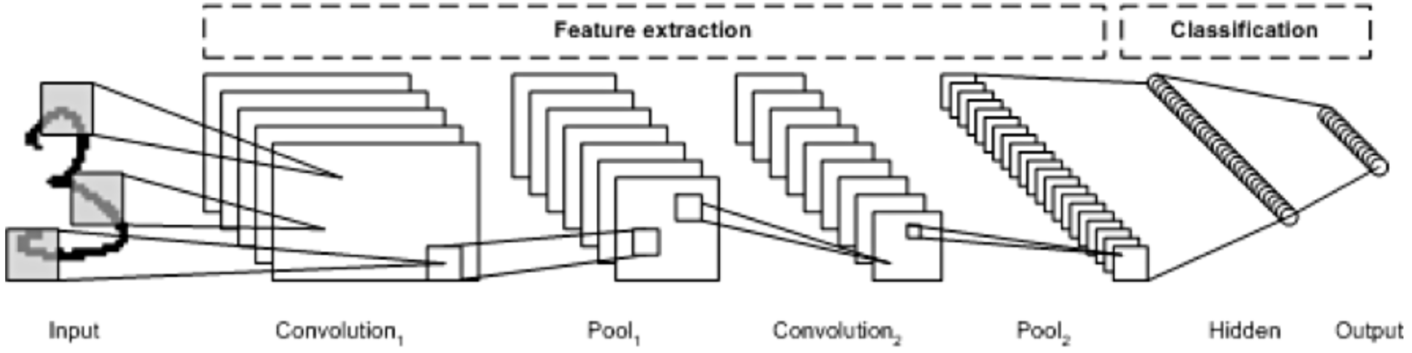
GCN

(Remind) Convolution Neural Network

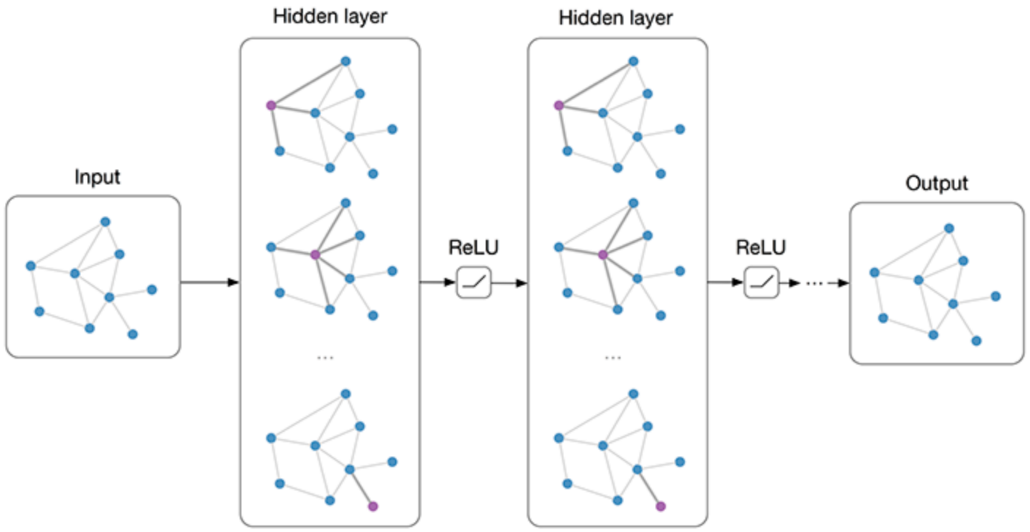
- CNN은 합성곱 연산을 사용하는 네트워크로 적합한 convolution 값을 학습한다.
- CNN은 각 레이어의 activation map의 값들을 업데이트 한다.

→ 이미지에 대한 CNN연산을 그래프에 대해 적용한다면? GCN

GCN



- 각 레이어를 거치면서 Convolution 연산을 수행하고 **픽셀 값**을 업데이트



- 각 레이어를 거치면서 Convolution 연산을 수행하고 **그래프 값**을 업데이트

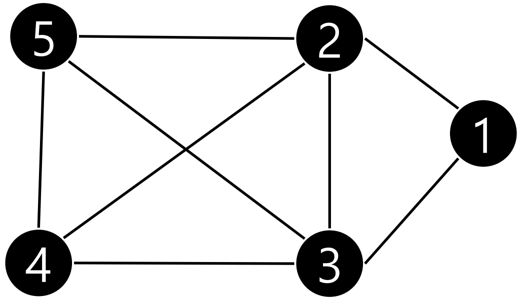
→ 노드의 feature 값 업데이트

GCN

Graph structure

Vertex(node) 정보 : Adjacency Matrix (A)

Edge 정보 : Node Feature Matrix (X)



Adjacency Matrix A (n x n)

	Node 1	Node 2	Node 3	Node 4	Node 5
Node 1	0	1	1	0	0
Node 2	1	0	1	1	1
Node 3	1	1	0	1	1
Node 4	0	1	1	0	1
Node 5	0	1	1	1	0

각각의 row : 노드의 연결 정보

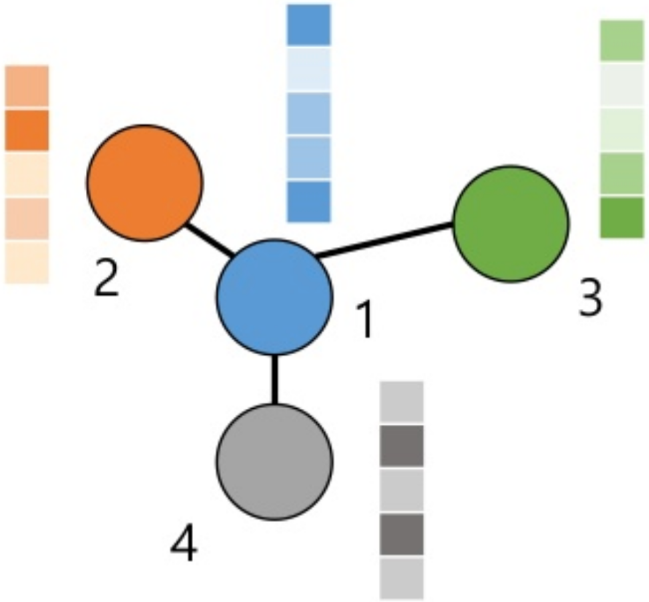
Node Feature Matrix X (n x f)

	Node 1	Node 2	Node 3	Node 4	Node 5
Node 1	1	1	1	0	0
Node 2	1	1	1	1	1
Node 3	1	1	1	1	1
Node 4	0	1	1	1	1
Node 5	0	1	1	1	1

각각의 row : 노드의 특징 벡터 정보

GCN

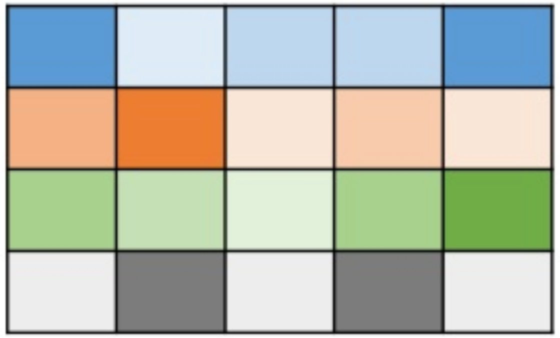
- Concept (Example)
  - 4 Nodes
  - 5 Features for each node



Adjacency Matrix **A** (4 x 4)

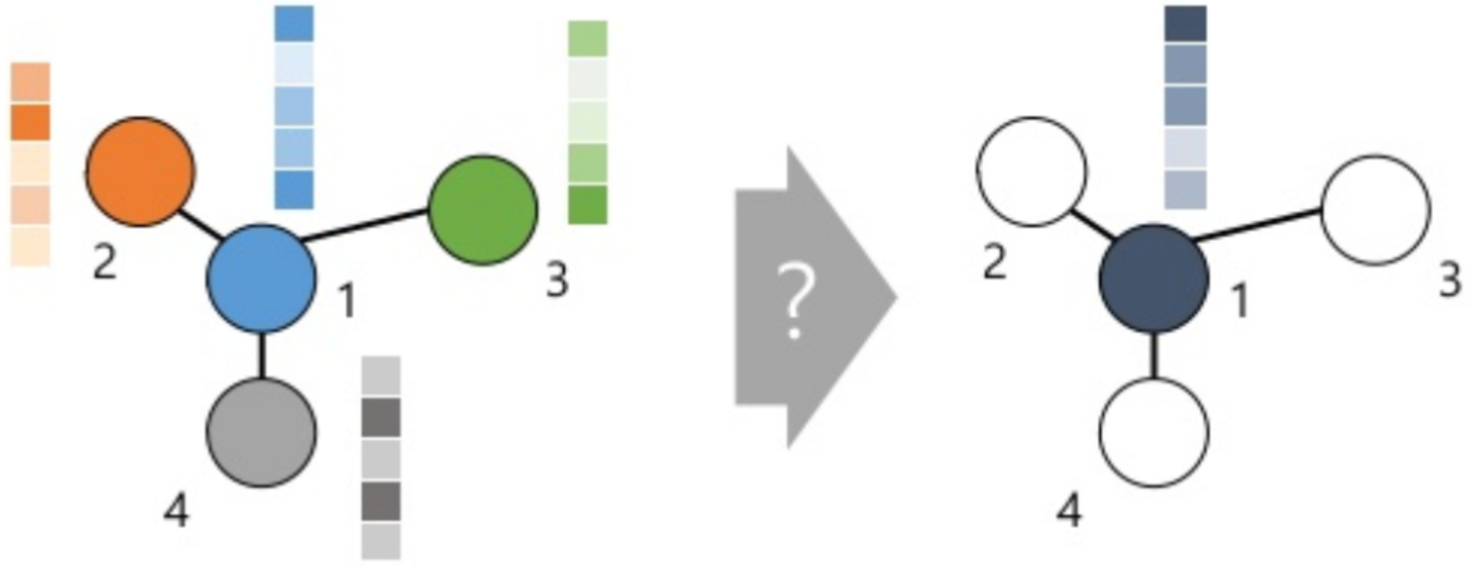
1	1	1	1
1	1	0	0
1	0	1	0
1	0	0	1

Feature Matrix **X** (4 x 5)



GCN

- Concept (Example)
  - Want to **update feature** values **+ Weight Sharing**
    + 주변 노드 정보 활용



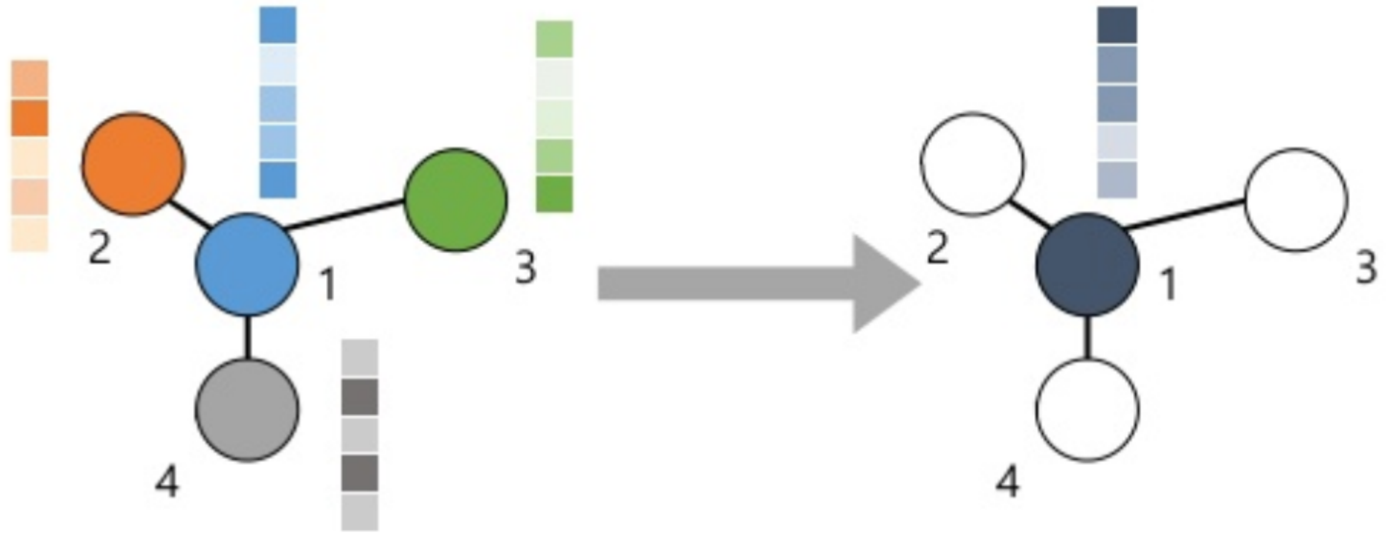
GCN

- Update node feature values

$H_i^{(l)}$: l번째 레이어의 i번째 노드 *feature*

$\sigma(\cdot)$: activation function

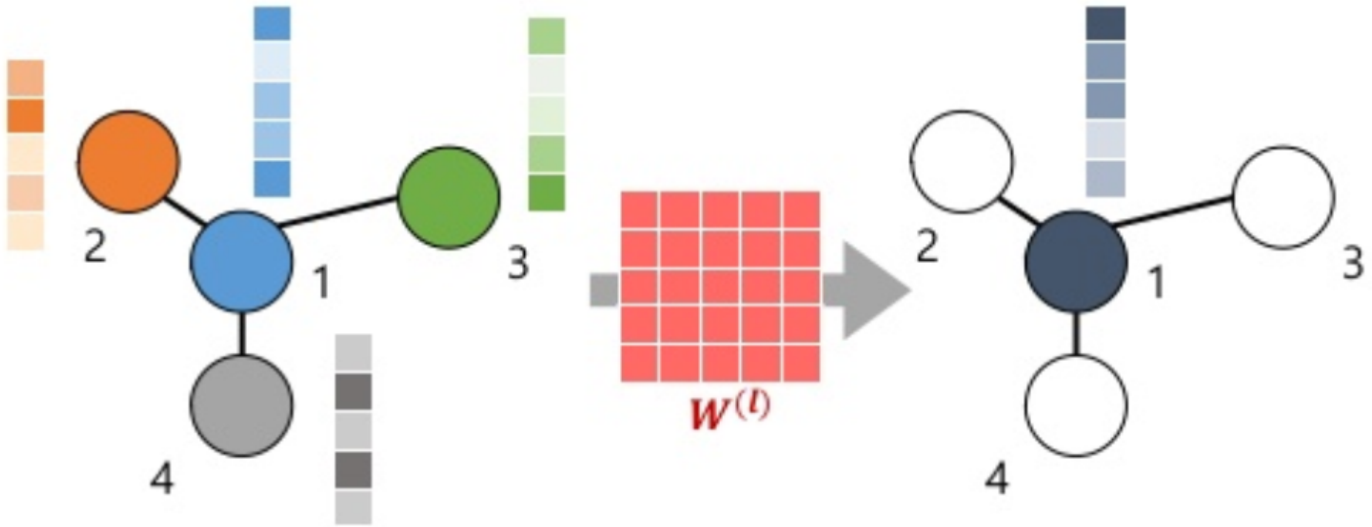
$W^{(l)}, b^{(l)}$: l번째 레이어의 weight 와 bias



$$H_1^{(l+1)} = \sigma\left(H_1^{(l)}W^{(l)} + H_2^{(l)}W^{(l)} + H_3^{(l)}W^{(l)} + H_4^{(l)}W^{(l)} + b^{(l)}\right)$$

GCN

- Update node feature values

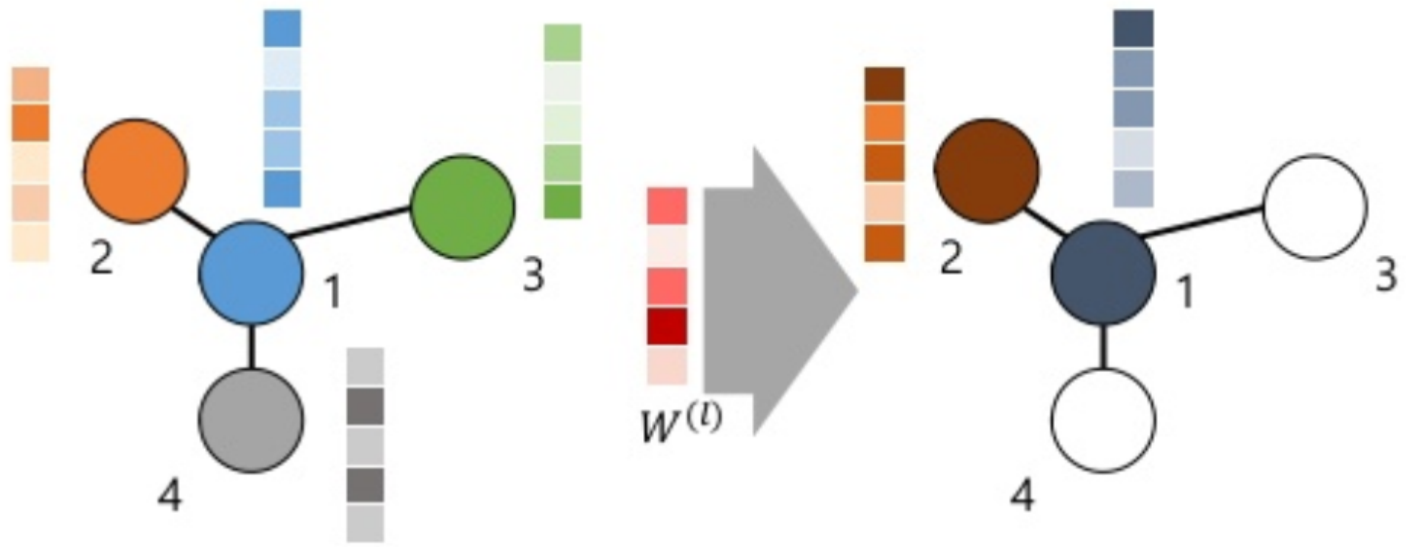


$$H_1^{(l+1)} = \sigma\left(H_1^{(l)}W^{(l)} + H_2^{(l)}W^{(l)} + H_3^{(l)}W^{(l)} + H_4^{(l)}W^{(l)} + b^{(l)}\right)$$

Weight Sharing

GCN

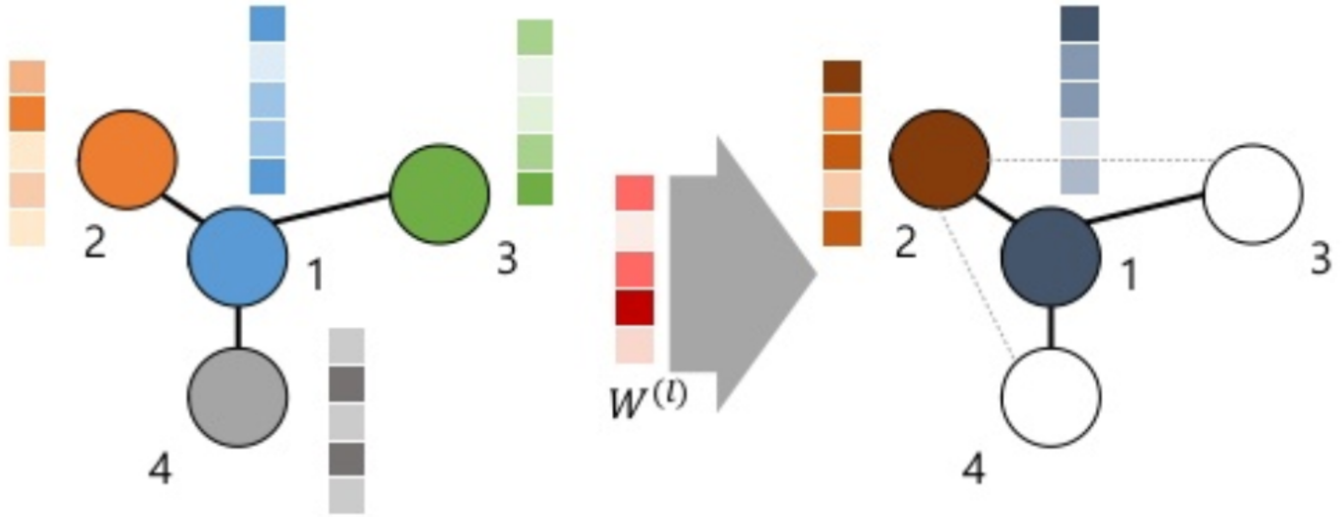
- Update node feature values



$$H_2^{(l+1)} = \sigma\left(H_1^{(l)}W^{(l)} + H_2^{(l)}W^{(l)} + H_3^{(l)}W^{(l)} + H_4^{(l)}W^{(l)} + b^{(l)}\right)$$

GCN

- Update node feature values

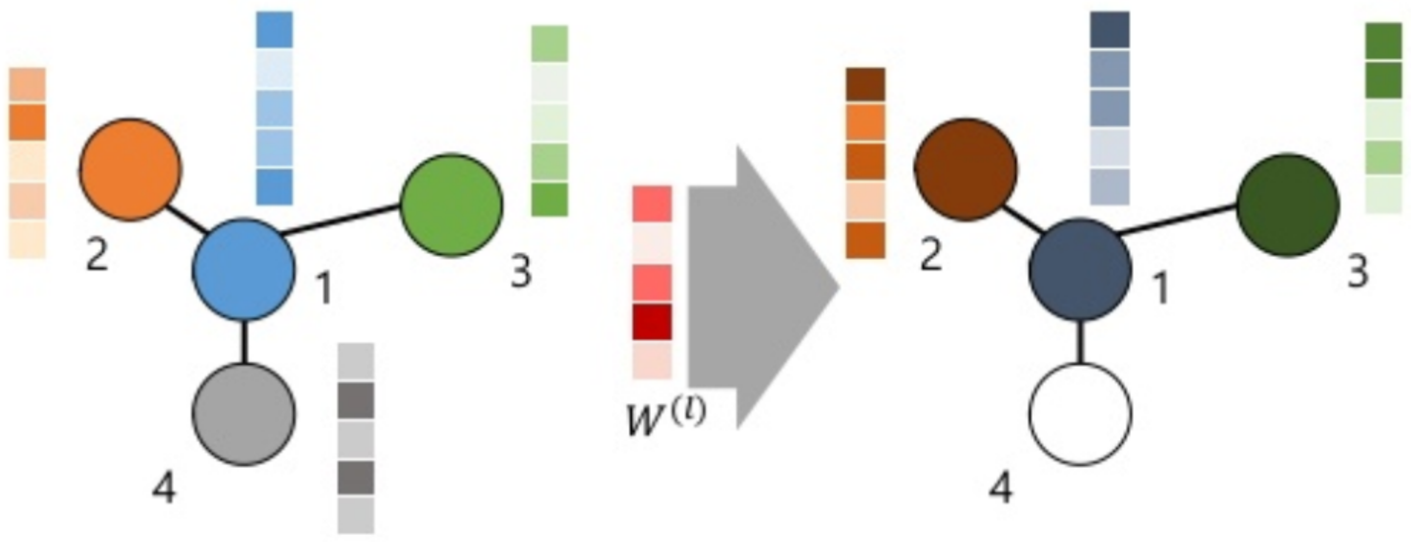


$$H_2^{(l+1)} = \sigma\left(H_1^{(l)}W^{(l)} + H_2^{(l)}W^{(l)} + \cancelto{0}{H_3^{(l)}}W^{(l)} + \cancelto{0}{H_4^{(l)}}W^{(l)} + b^{(l)}\right)$$

주변 노드 정보 활용

GCN

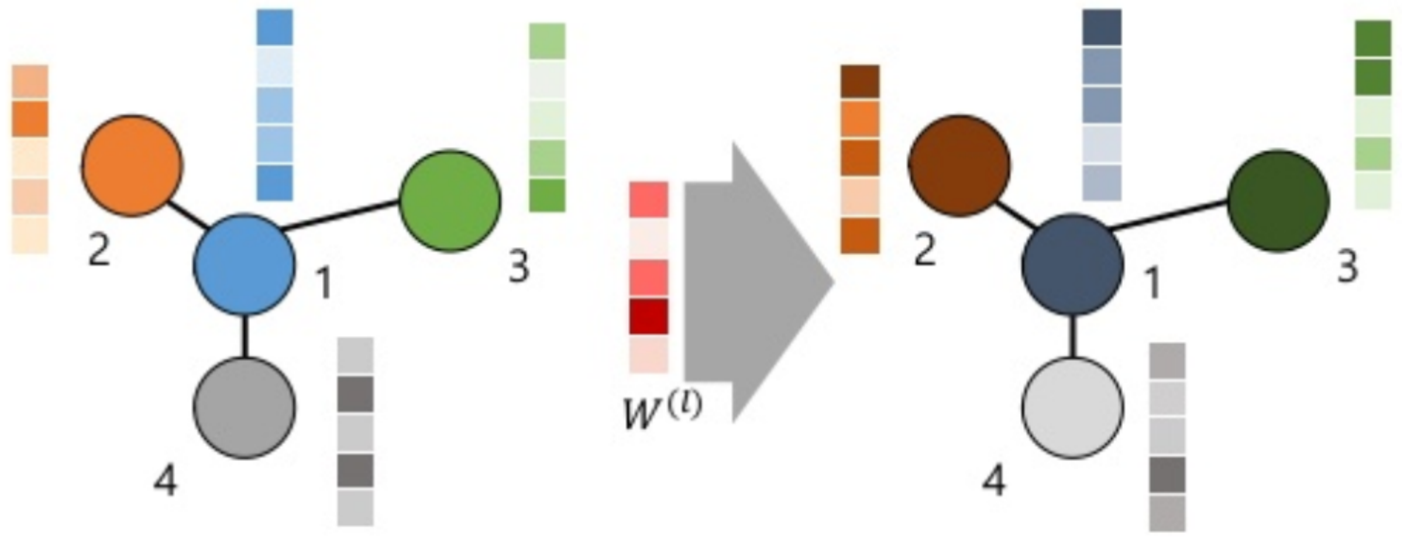
- Update node feature values



$$H_3^{(l+1)} = \sigma\left(H_1^{(l)}W^{(l)} + H_3^{(l)}W^{(l)} + b^{(l)}\right)$$

GCN

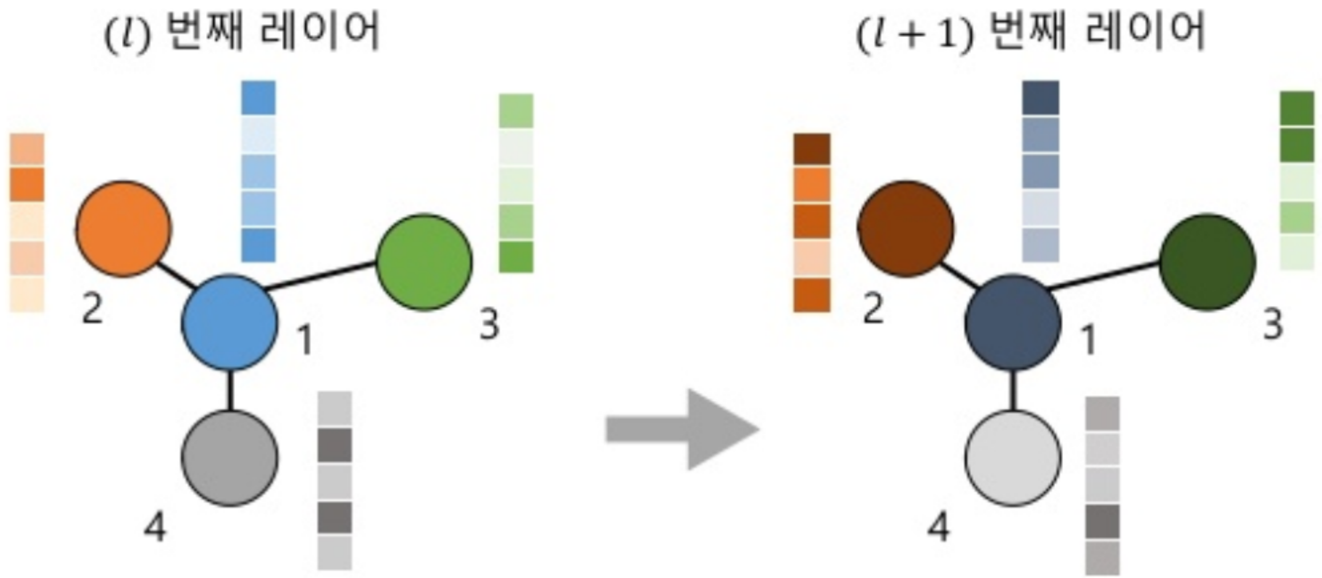
- Update node feature values



$$H_4^{(l+1)} = \sigma\left(H_1^{(l)}W^{(l)} + H_4^{(l)}W^{(l)} + b^{(l)}\right)$$

GCN

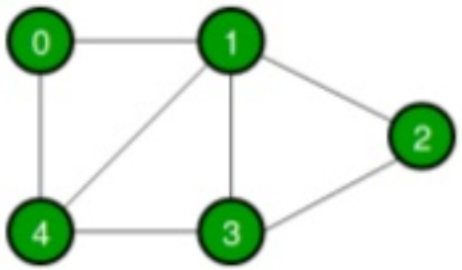
- Update node feature values



$$H_i^{(l+1)} = \sigma\left(\sum_{j\in N(i)} H_j^{(l)} W^{(l)} + b^{(l)}\right) \quad \text{or} \quad H^{(l+1)} = \sigma\left(AH^{(l)}W^{(l)} + b^{(l)}\right)$$

GCN

- How to update hidden states in GCN (Matrix 연산으로 이해하기)

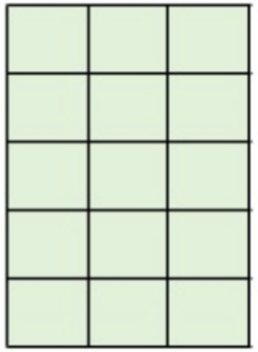


Adjacency Matrix
A (5 x 5)

1	1	0	0	1
1	1	1	1	1
0	1	1	1	0
0	1	1	1	1
1	1	0	1	1

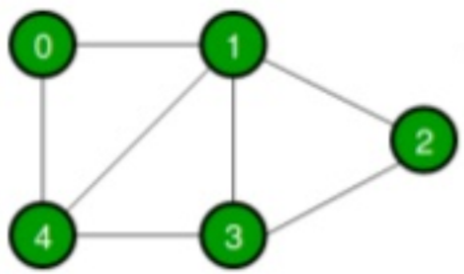
각 노드는 3개의
feature 값을 갖는다
(예시) 이름 성별 나이

Feature Matrix
$X = H^0$ (5 x 3)



GCN

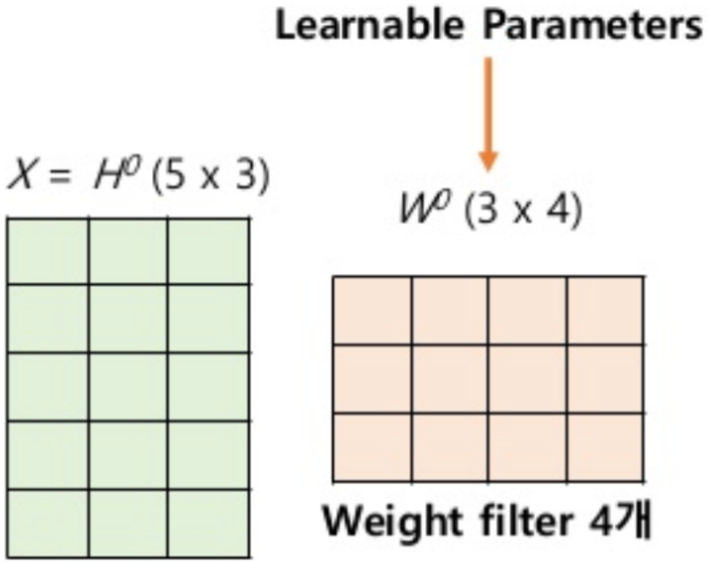
- How to update hidden states in GCN (Matrix 연산으로 이해하기)



$$H^{(l+1)} = \sigma\left(\boldsymbol{A}\boldsymbol{H}^{(l)}\boldsymbol{W}^{(l)} + b^{(l)}\right)$$

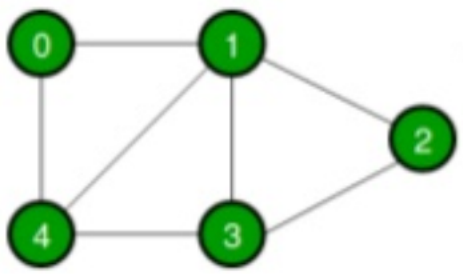
A (5 x 5)

1	1	0	0	1
1	1	1	1	1
0	1	1	1	0
0	1	1	1	1
1	1	0	1	1

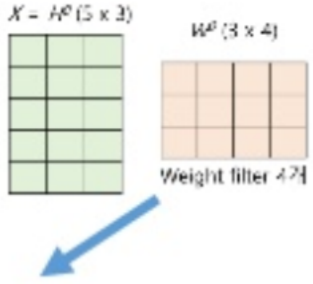


GCN

- How to update hidden states in GCN (Matrix 연산으로 이해하기)



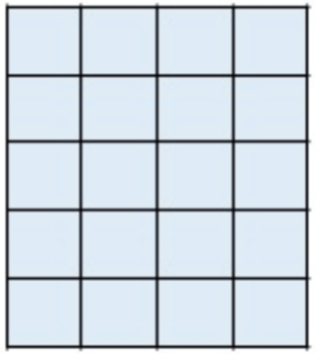
$$H^{(l+1)} = \sigma\left(AH^{(l)}W^{(l)} + b^{(l)}\right)$$



A (5 x 5)

1	1	0	0	1
1	1	1	1	1
0	1	1	1	0
0	1	1	1	1
1	1	0	1	1

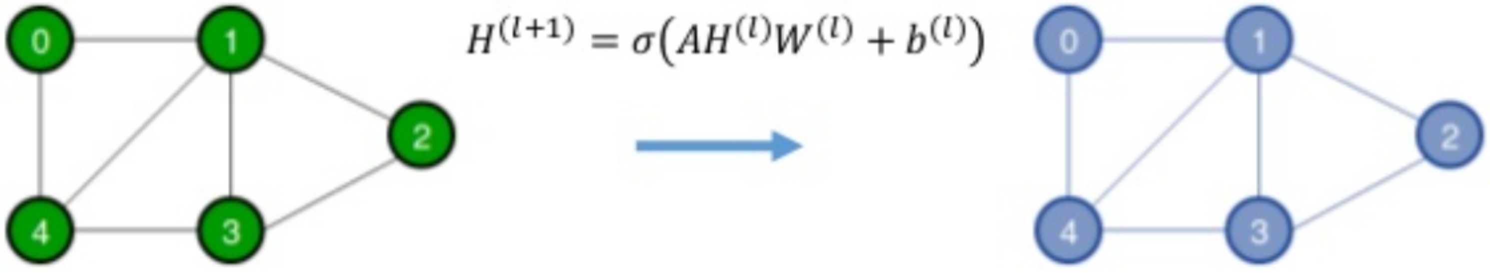
$X * W^0$ (5 x 4)

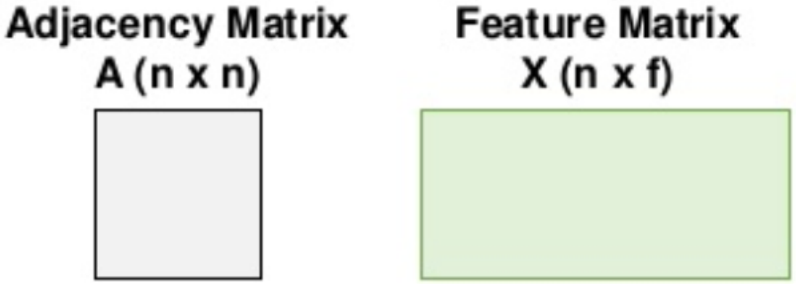


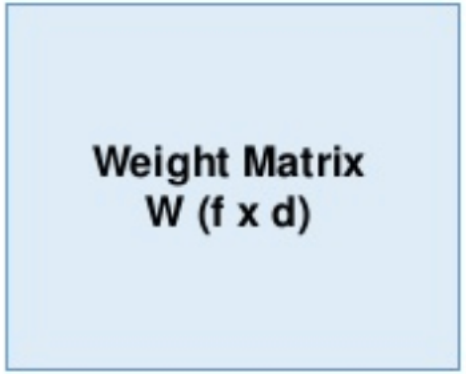
GCN

- How to update hidden states in GCN (Matrix 연산으로 이해하기)

$$H^{(l+1)} = \sigma\left(AH^{(l)}W^{(l)} + b^{(l)}\right)$$



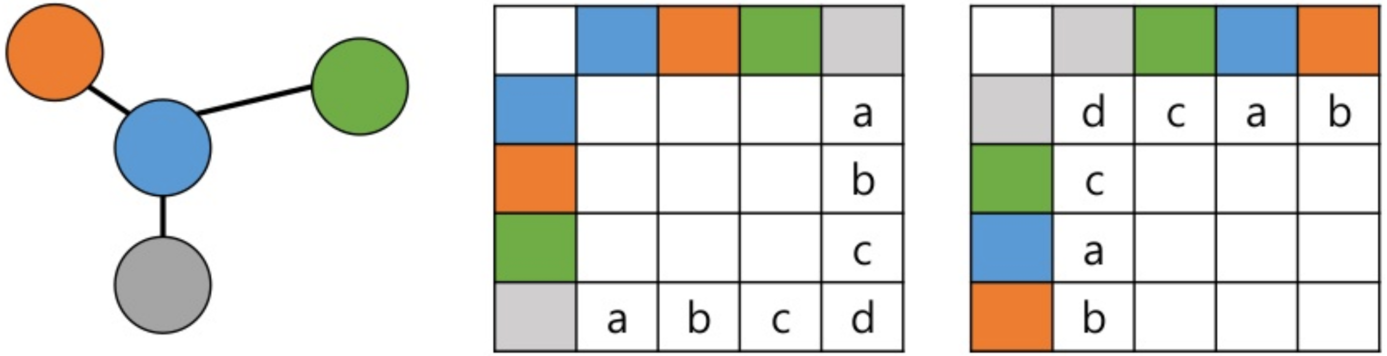




(참고) n: 노드의 수 / f: 특징 수 / d: 필터수 (depth)
결국 d가 다음 레이어의 f 값이 된다

GCN

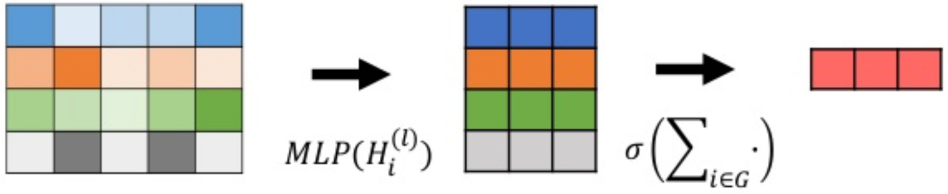
- Readout: Permutation Invariance



노드 순서에 따라 값의 변동이 있을 수 있다

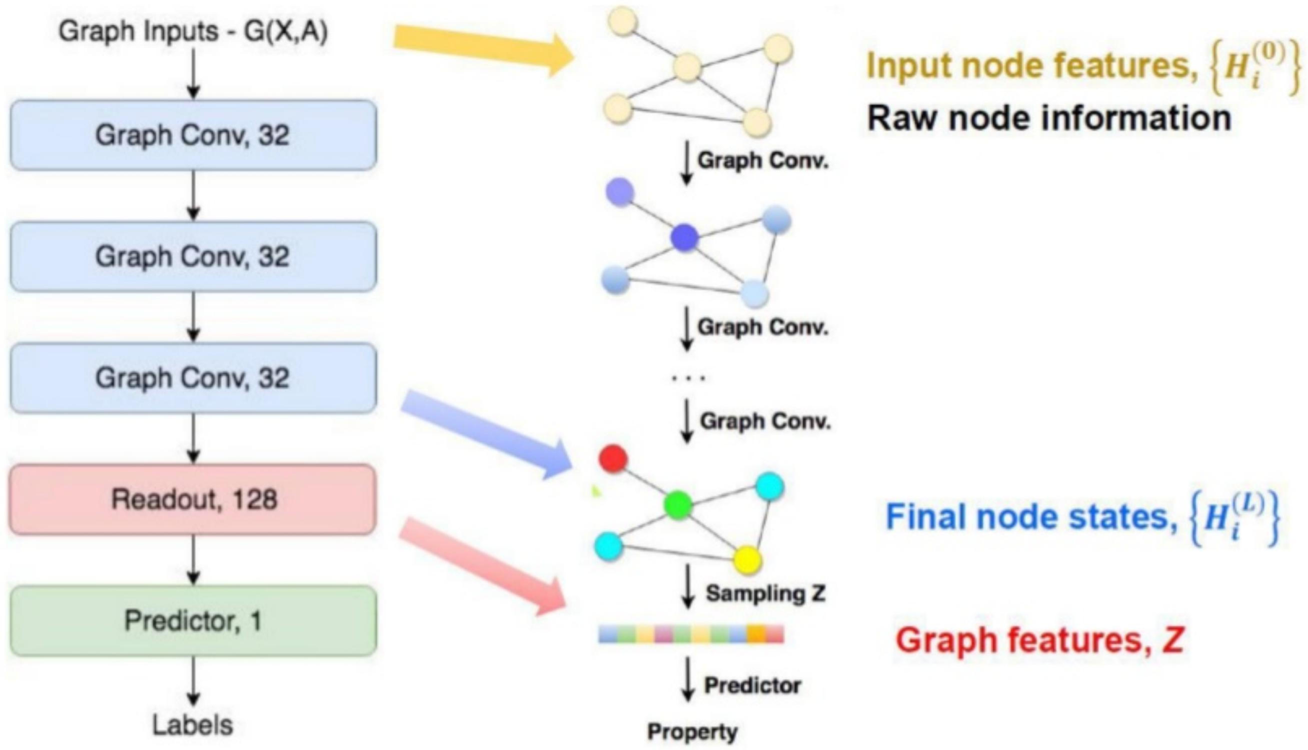
Readout 방법 중 한가지: Node-wise summation

$$z_G = \sigma\left(\sum_{i \in G} MLP(H_i^{(l)})\right)$$



GCN

- Overall Structure of GCN



Methodology

Methodology

크게 3가지 부분으로 나뉨

1. **Sequential Context Encoding**
2. **Speaker-Level Context Encoding**
3. **Classification**

Methodology

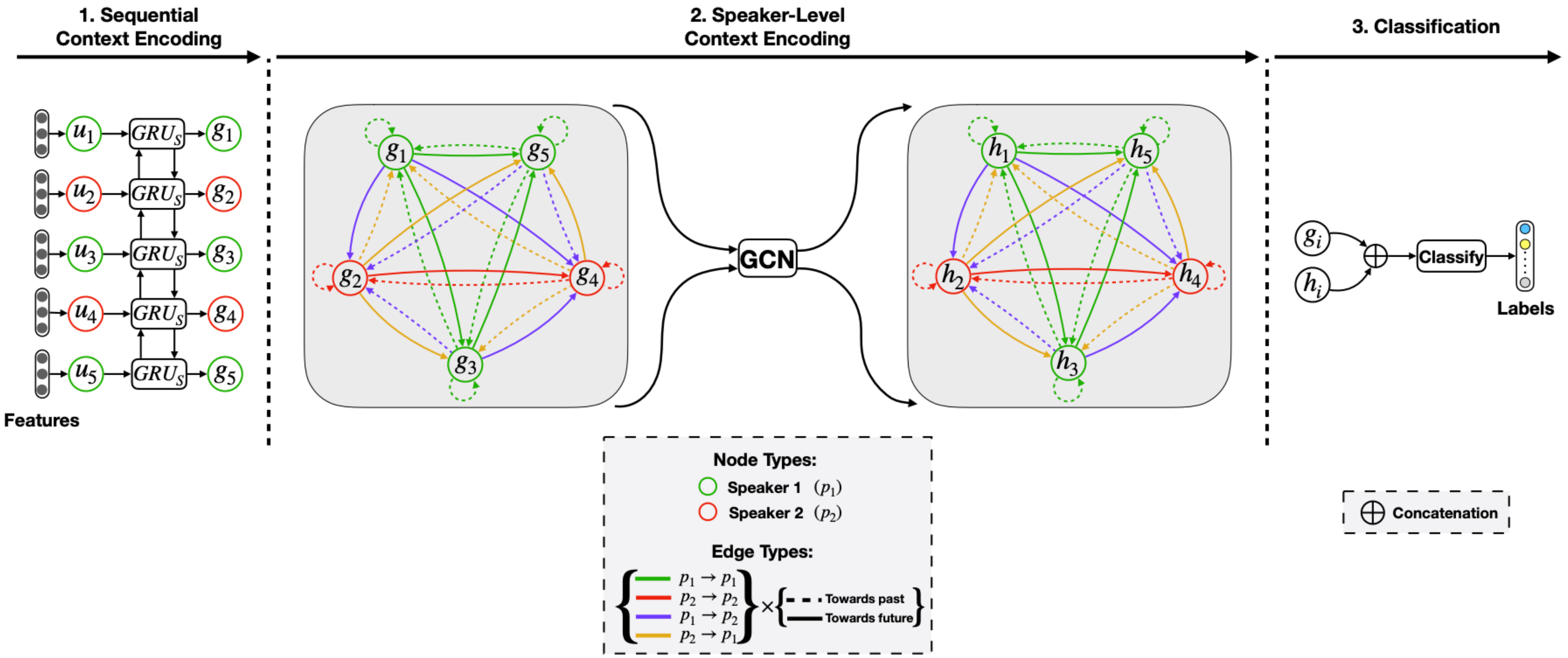
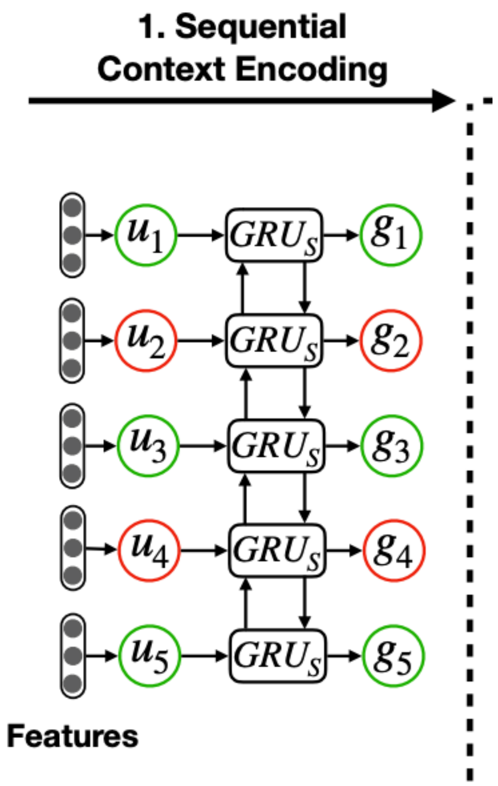


Figure 3: Overview of DialogueGCN, congruent to the illustration in Table 1.

Methodology

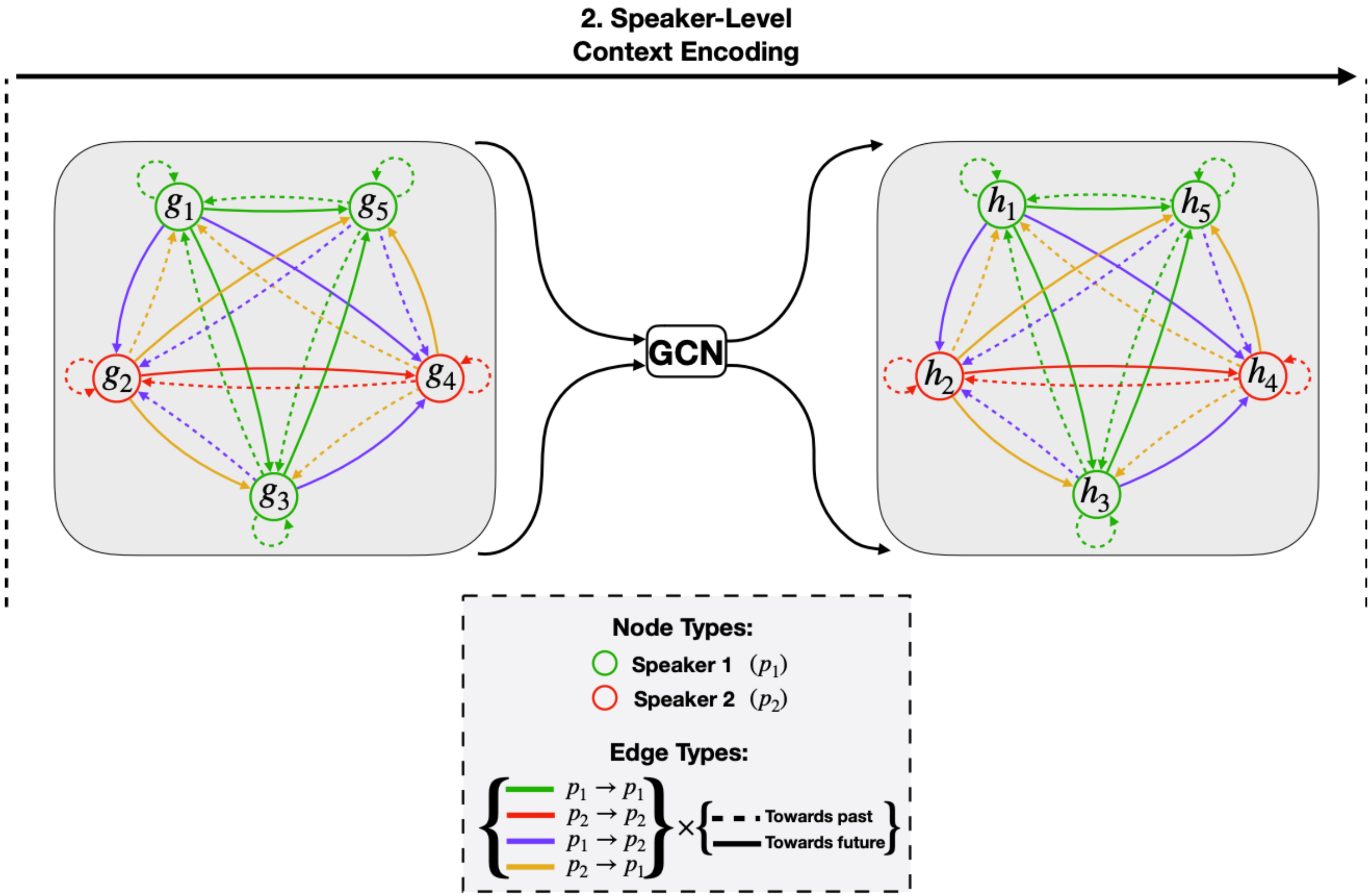
Sequential Context Encoding



$$g_i = \overleftrightarrow{GRU_S}(g_{i(+,-)1}, u_i)$$

Methodology

Speaker-Level Context Encoding



Methodology

Speaker-Level Context Encoding

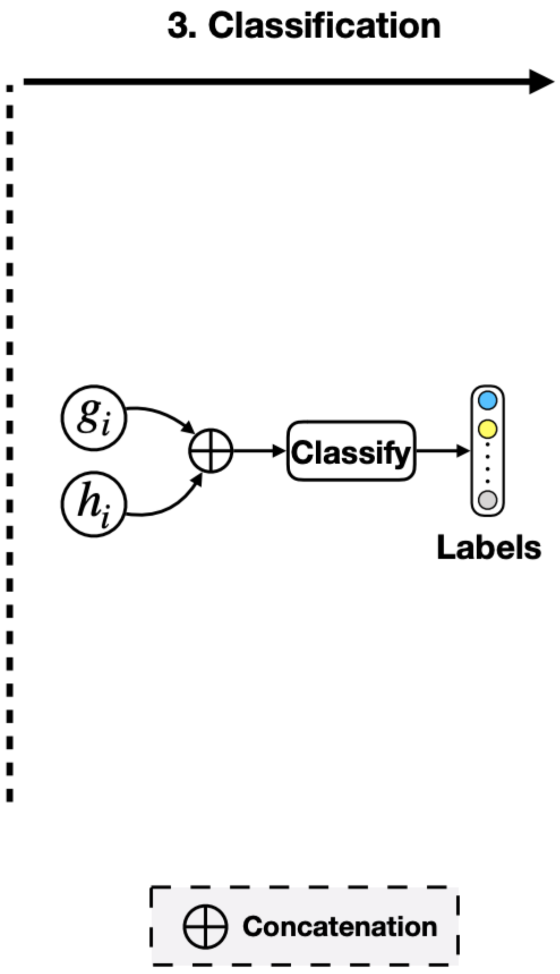
Relation	$p_s(u_i), p_s(u_j)$	$i < j$	(i, j)
1	p_1, p_1	Yes	(1,3), (1,5), (3,5)
2	p_1, p_1	No	(1,1), (3,1), (3,3) (5,1), (5,3), (5,5)
3	p_2, p_2	Yes	(2,4)
4	p_2, p_2	No	(2,2), (4,2), (4, 4)
5	p_1, p_2	Yes	(1,2), (1,4), (3,4)
6	p_1, p_2	No	(3,2), (5,2), (5,4)
7	p_2, p_1	Yes	(2,3), (2,5), (4,5)
8	p_2, p_1	No	(2,1), (4,1), (4,3)

Table 1: $p_s(u_i)$ and $p_s(u_j)$ denotes the speaker of utterances u_i and u_j. 2 distinct speakers in the conversation implies $2 * M^2 = 2 * 2^2 = 8$ distinct relation types. The rightmost column denotes the indices of the vertices of the constituting edge which has the relation type indicated by the leftmost column.

$$\alpha_{ij} = \text{softmax}(g_i^T W_e[g_{i-p}, \ldots, g_{i+f}]), \quad \text{for } j = i-p, \ldots, i+f. \tag{1}$$

Methodology

Speaker-Level Context Encoding



$$h_i = [g_i, h_i^{(2)}], \quad (4)$$

$$\beta_i = \text{softmax}(h_i^T W_\beta [h_1, h_2 \ldots, h_N]), \quad (5)$$

$$\tilde{h}_i = \beta_i [h_1, h_2, \ldots, h_N]^T. \quad (6)$$

$$l_i = \text{ReLU}(W_l \tilde{h}_i + b_l), \quad (7)$$

$$\mathcal{P}_i = \text{softmax}(W_{smax} l_i + b_{smax}), \quad (8)$$

$$\hat{y}_i = \underset{k}{\text{argmax}}(\mathcal{P}_i[k]). \quad (9)$$

Experiments

Experiments
Dataset used

Dataset	# dialogues			# utterances		
	train	val	test	train	val	test
IEMOCAP	120		31	5810		1623
AVEC	63		32	4368		1430
MELD	1039	114	280	9989	1109	2610

Table 2: Training, validation and test data distribution in the datasets. No predefined train/val split is provided in IEMOCAP and AVEC, hence we use 10% of the training dialogues as validation split.

Experiments
Results

Methods	IEMOCAP													
	Happy		Sad		Neutral		Angry		Excited		Frustrated		**Average(w)**	
	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1
CNN	27.77	29.86	57.14	53.83	34.33	40.14	61.17	52.44	46.15	50.09	62.99	55.75	48.92	48.18
Memnet	25.72	33.53	55.53	61.77	58.12	52.84	59.32	55.39	51.50	58.30	67.20	59.00	55.72	55.10
bc-LSTM	29.17	34.43	57.14	60.87	54.17	51.81	57.06	56.73	51.17	57.95	67.19	58.92	55.21	54.95
bc-LSTM+Att	30.56	35.63	56.73	62.90	57.55	53.00	59.41	59.24	52.84	58.85	65.88	59.41	56.32	56.19
CMN	25.00	30.38	55.92	62.41	52.86	52.39	61.76	59.83	55.52	60.25	71.13	60.69	56.56	56.13
ICON	22.22	29.91	58.78	64.57	62.76	57.38	64.71	63.04	58.86	63.42	67.19	60.81	59.09	58.54
DialogueRNN	25.69	33.18	75.10	78.80	58.59	59.21	64.71	**65.28**	80.27	**71.86**	61.15	58.91	63.40	62.75
DialogueGCN	40.62	**42.75**	89.14	**84.54**	61.92	**63.54**	67.53	64.19	65.46	63.08	64.18	**66.99**	65.25	**64.18**

Table 3: Comparison with the baseline methods on IEMOCAP dataset; Acc. = Accuracy; bold font denotes the best performances. Average(w) = Weighted average.

Experiments
Results

Methods	AVEC				MELD
	Valence	Arousal	Expectancy	Power	
CNN	0.545	0.542	0.605	8.71	55.02
Memnet	0.202	0.211	0.216	8.97	-
bc-LSTM	0.194	0.212	0.201	8.90	56.44
bc-LSTM+Att	0.189	0.213	0.190	8.67	56.70
CMN	0.192	0.213	0.195	8.74	-
ICON	0.180	0.190	0.180	8.45	-
DialogueRNN	0.168	0.165	0.175	7.90	57.03
DialogueGCN	**0.157**	**0.161**	**0.168**	**7.68**	**58.10**

Table 4: Comparison with the baseline methods on AVEC and MELD dataset; MAE and F1 metrics are user for AVEC and MELD, respectively.

Experiments
Results

Sequential Encoder	Speaker-Level Encoder	F1
✓	✓	64.18
✓	✗	55.30
✗	✓	56.71
✗	✗	36.75

Table 5: Ablation results w.r.t the contextual encoder modules on IEMOCAP dataset.

Speaker Dependency Edges	Temporal Dependency Edges	F1
✓	✓	64.18
✓	✗	62.52
✗	✓	61.03
✗	✗	60.11

Table 6: Ablation results w.r.t the edge relations in speaker-level encoder module on IEMOCAP dataset.

Conclusion & Discussion

Conclusion & Discussion

Conclusion

DialogueGCN 제안했는데, **우리 모델 3개 benchmark dataset에서 전부 SOTA임!**

Conclusion & Discussion

Conclusion

응. 근데 지금은 아님^^7 (CESTa, ACL 2020)

DialogueGCN 제안했는데, **우리 모델 3개 benchmark dataset에서 전부 SOTA임!**

Models	IEMOCAP							DailyDialogue	MELD
	Happy	Sad	Neutral	Angry	Excited	Frustrated	Avg.(w)	Avg.(micro)	Avg.(w)
CNN	35.34	53.66	51.61	62.17	50.66	55.56	51.28	49.27	55.86
CNN+cLSTM	33.90	69.76	48.40	57.55	62.37	57.64	56.04	51.84	56.87
DialogueRNN	37.94	78.08	58.95	**64.86**	68.11	58.85	62.26	51.64	57.07
DialogueGCN	42.75	**84.54**	63.54	64.19	63.08	**66.99**	64.18	-	58.10
KET	-	-	-	-	-	-	59.56	53.37	58.18
CESTa	**47.70**	80.82	**64.76**	63.41	**75.95**	62.65	**67.10**	**63.12**	**58.36**

Table 2: Comparisons with baselines and state-of-the-art methods. Best performances are highlighted in bold.

Conclusion & Discussion
Conclusion

Transformer는 못이김..ㅎㅎ

응. 근데 지금은 아님^^7 (CESTa, ACL 2020)

DialogueGCN 제안했는데, **우리 모델 3개 benchmark dataset에**

Models	IEMOCAP							Dai	
	Happy	Sad	Neutral	Angry	Excited	Frustrated	Avg.(w)	A	
CNN	35.34	53.66	51.61	62.17	50.66	55.56	51.28		
CNN+cLSTM	33.90	69.76	48.40	57.55	62.37	57.64	56.04		
DialogueRNN	37.94	78.08	58.95	**64.86**	68.11	58.85	62.26		
DialogueGCN	42.75	**84.54**	63.54	64.19	63.08	**66.99**	64.18		
KET	-	-	-	-	-	-	59.56	[illegible]	[illegible]
CESTa	**47.70**	80.82	**64.76**	63.41	**75.95**	62.65	**67.10**	**63.12**	**58.36**

Table 2: Comparisons with baselines and state-of-the-art methods. Best performances are highlighted in bold.

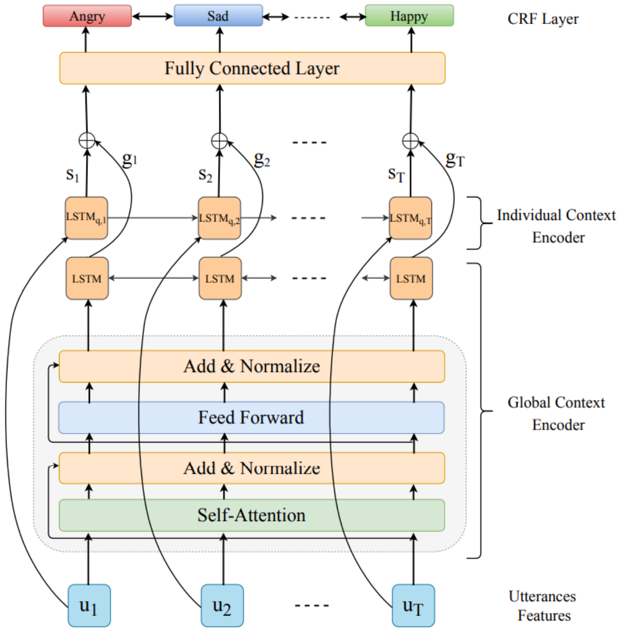


Figure 2: Overview of CESTa. The Transformer-enhanced global context encoder takes the textual feature $\mathbf{u}_t$ of the tth utterance in a conversation as input and produces encoding $\mathbf{g}_t$. Also, $\mathbf{u}_t$ is fed into the individual context encoder to update states for the corresponding speaker of which index is $q = q(\mathbf{u}_t)$ and outputs another encoding $\mathbf{s}_t$. A CRF layer is applied over the concatenation of each $\mathbf{g}_t$ and $\mathbf{s}_t$ to obtain the final prediction for each utterance in the conversation.

Conclusion & Discussion
Discussion

논문을 읽고 든 생각!

오~ GCN을 emotion recognition에 처음으로 적용 했다니 신선했다!

Conclusion & Discussion
Discussion

논문을 읽고 든 생각!

그런데 화자 수를 고정해야 한다니 너무 아쉬운데..? 화자 수가 바뀌는 것에 대한 처리는 못하는 거임?

Conclusion & Discussion
Discussion

논문을 읽고 든 생각!

그런데 화자 수를 고정해야 한다니 너무 아쉬운데..? 화자 수가 바뀌는 것에 대한 처리는 못하는 거임?

→ 이게 제 연구 주제 입니다^^ 이거 논문 쓰느라 급하게 논문 바꿨어요…

Thank you.

Appendix

Appendix
references

Paper

DialogueGCN : https://www.aclweb.org/anthology/D19-1015.pdf

Etc.

https://www.slideshare.net/DonghyeonKim7/graph-convolutional-network-gcn