요약

- 이전 논문들 (ST-GCN, 2S-AGCN)의 단점을 지적하면서(표현의 한계점, 남용) 최적화된 graph convolution layer을 제안함
- 2S-AGCN에서 AGC-Convs 영역 변경과 Learnable Adjacency Matrix의 초기값 설정에 관련된 제안이 주된 내용
    - 2S-AGCN의 AGC에서 Adjacency Matrix A,B,C 를 분리하고 B,C matrix에 대해서만 learnable 하게 설정
      B : Convolution Layer에 저장되는 Adjacency Matrix. C : 입력 feature를 가지고 Attention한 Adjacency Matrix
    - 어떤 초기값 설정(+ labeling)이 좋은지 실험을 통해서 검증

제안 기법

$$\mathbf{f}_{out} = \sum_{k}^{K_v} \mathbf{W}_k \mathbf{f}_{in} \mathbf{A}_k \circ \mathbf{M}_k, \quad (2)$$

$$\mathbf{f}_{out} = \sum_{k}^{K_v} \mathbf{W}_k \mathbf{f}_{in} (\mathbf{A}_k + \mathbf{B}_k + \mathbf{C}_k), \quad (3)$$

$$FL(\mathbf{W}_k, \mathbf{f}_{in})_{(i,k)} = \sum_{m}^{M} \mathbf{W}_k(m) \cdot \mathbf{f}_{in}(v_i, m),$$

$$VF(\mathbf{L}, \mathbf{f}_{in})_{(i)} = \sum_{k}^{K_v} \sum_{j}^{N} \mathbf{L}_{ij}^{k} \circ \mathbf{f}_{in}^{k}(v_j),$$

$$GraphConv(\mathbf{W}, \mathbf{L}, \mathbf{f}_{in})_{(i)} = VF_{(i)}(\mathbf{L}, FL_{(i,k)}(\mathbf{W}, \mathbf{f}_{in})), \quad (4)$$

$$\mathbf{L} = \mathbf{A}_0 \cup \cdots \cup \mathbf{A}_d. \quad (5)$$

- 그림 (1)에서 (c)가 본 논문의 제안 기법 (식 (4))
  - 다른 논문에서 Attention / Adjacency Matrix를 다양하게 표현
    ST-GCN (a) = 식 (2) , 2s-AGCN (b) = 식 (3)
  - 본 논문에서는 learnable Adjacency Matrix L_k [Kernel Size, Joint Num, Joint Num] 간소화
- L_k를 초기화 방법을 4개 제안하고 실험을 통해 제일 좋은 초기화 기법을 찾음
  - a) One Partition (식 5)
    - ST-GCN의 Uni-Labeling과 동일한 기법
    - Label(=kernel size)를 단일로 통일하고 거리 d에 따라 물리적인 연결로 adjacency matrix를 표현
  - b) Multipartition
    - ST-GCN의 Distance Partitioning과 동일한 기법
    - 거리 d만큼 물리적인 연결로 joint별 adjacency matrix를 표현한 뒤, 거리의 정도에 따라 다른 label을 적용하는 방법
    - ex) d=2 일때, labels={0,1,2} 및 식 (5)로 adjacency matrix 표현
  - c) Weight Normalization
    - (b) 방식에 ST-GCN의 Normalization 활용하기
    - joint 별 edge개수로 평균을 내는 방식으로 normalize function으로 softmax/Relu를 실험에 사용
  - d) Additional Bias
    - ST-GCN의 Spatial Configuration에(=2s AGCN의 A matrix) Bias를 추가하는 방식
    - 물리적인 연결이 Action Recognition에서 최적은 아닐지라도 중요한 요소이므로 그대로 채용하자는 뜻

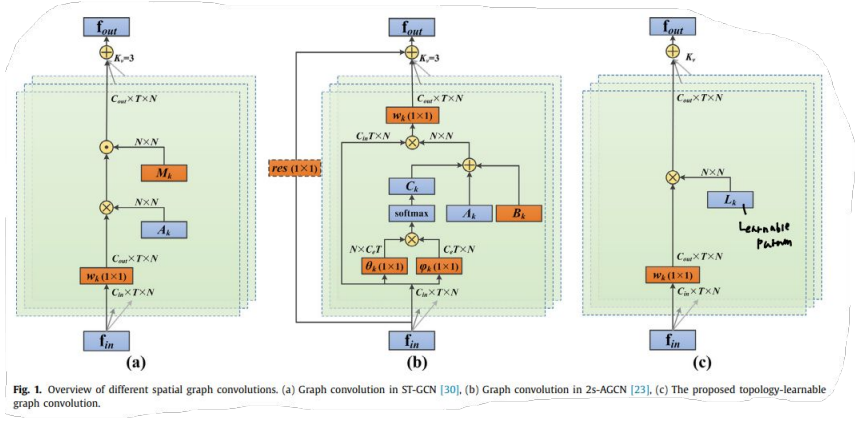

Fig. 1. Overview of different spatial graph convolutions. (a) Graph convolution in ST-GCN [30], (b) Graph convolution in 2s-AGCN [23], (c) The proposed topology-learnable graph convolution.

실험결과

- 표 1) 논문이 제안한 adjacency matrix 초기화 방법에 따른 성능 비교
  - Bias를 추가하는 (d) 방법이 제일 성능이 잘 나오는 것을 확인
- 표 2) 기존 논문의 SOTA(2s-AGCN)와 본 논문에서 제안한 기법을 제거하면서 성능변화 관찰
  - SepTConv의 경우, Depth-wise Convolution을 시간축을 기준으로 적용한 것
- 표 3, 4) 다른 논문들과 비교

Table 1
Evaluation of different initialization strategies on the Kinetics-Skeleton dataset based on ST-GCN and on the X-View subset of NTU RGB+D. The symbol * means that it is different from the distance partitioning in [30] since the adjacency matrices are merged into one matrix when K_v=1 in this work. Our implementation is based on the released code of ST-GCN [30] and 2s-AGCN [23].

Cases	Initialization Strategy	Kinetics-Skeleton Top-1(%)	Kinetics-Skeleton Top-5(%)	NTU RGB+D Top-1(%)
-	ST-GCN (Distance Partitioning) [30]	29.10	51.30	–
	ST-GCN (Spatial Partitioning) [23,30]	29.90	52.20	91.10
	ST-GCN (Spatial Partitioning) + Edge Importance Weighting [23,30]	30.70	52.80	92.70
	AGCN (Spatial Partitioning) wo/A [23]	–	–	93.40
	AGCN (Spatial Partitioning) [23]	–	–	93.70
(a)	$K_v = 1$ Distance Partitioning($d = 1$)*	30.33	52.65	93.53
	$K_v = 1$ Distance Partitioning($d = 2$)*	30.40	52.46	93.56
(b)	$K_v = 2$ Distance Partitioning($d = 1$)	30.99	53.16	93.89
	$K_v = 3$ Distance Partitioning($d = 2$)	30.70	52.49	93.73
(c)	$K_v = 2$ Distance Partitioning($d = 1$) + Relu_Norm	30.79	53.60	93.81
	$K_v = 2$ Distance Partitioning($d = 1$) + Softmax_Norm	30.17	53.16	93.24
(d)	$K_v = 2$ Distance Partitioning($d = 1$) + Bias	32.05	54.06	93.98
	$K_v = 3$ Spatial Partitioning + Bias	**32.09**	**54.09**	**94.01**

Table 2
Effectiveness and efficiency on the X-View subset of NTU RGB+D based on 2s-AGCN. The TL-GCN means the proposed topology-learnable graph convolution networks using the spatial partitioning and the constant bias.

Network	Stream	Accuracy(%)	Parameter	Computation
2s-AGCN	Joint	93.70	3.33M	518 s
	Bone	93.20		
	Both	95.10		
TL-GCN w/o Bias	Joint	93.92	3.01M	360 s
	Bone	93.87		
	Both	95.17		
TL-GCN w/ Bias	Joint	94.01	3.01M	360 s
	Bone	93.99		
	Both	95.40		
TL-GCN w/ Bias + SepTConv	Joint	93.52	0.73M	286 s
	Bone	93.18		
	Both	94.99		

Table 3
Comparisons of the validation accuracy with state-of-the- art methods on the NTU RGB+D dataset.

Methods	X-Sub(%)	X-View(%)
Deep LSTM [20]	60.7	67.3
VA-LSTM [31]	79.2	87.8
Ind-RNN [16]	81.8	88.0
TCN [10]	74.3	83.1
Clips+CNN+MTLN [9]	79.6	84.8
Synthesized CNN [17]	80.0	87.2
CNN+Motion+Trans [14]	83.2	89.3
3Scale ResNet152 [12]	85.0	92.3
PA-GCN [19]	80.4	82.7
ST-GCN [30]	81.5	88.3
DPRL+GCNN [25]	83.5	89.8
BPLHM [32]	85.4	91.1
3s RA-GCN [24]	85.9	93.5
AS-GCN [15]	86.8	94.2
PB-GCN [26]	87.5	93.2
2s-ASGCN [21]	88.3	95.4
2s-AGCN [23]	88.5	95.1
Two-stream TL-GCN	**89.2**	**95.4**

Table 4
Comparisons of the validation accuracy with state-of-the-art methods on the Kinetics-Skeleton dataset.

Methods	Top-1(%)	Top-5(%)
Feature Enc[3].	14.9	25.8
Deep LSTM [20]	16.4	35.3
TCN [10]	20.3	40.0
ST-GCN [30]	30.7	52.8
BPLHM [32]	33.4	56.2
2s-ASGCN [21]	34.5	56.9
AS-GCN [15]	34.8	56.5
2s-AGCN [23]	36.1	58.7
Two-stream TL-GCN	**36.2**	**59.0**