Temporal Convolutional Attention-based Network For Sequence Modeling

https://arxiv.org/pdf/2002.12530

0. Introduction

- TCN: RNN/LSTM 단점 보완하려고 나온 시퀀스 모델
- Conv 기반 → 순차 처리 불필요, 병렬 처리 가능
- Long-range dependency 처리 가능 → dilated convolution 사용
- Causal 구조 → 미래 정보 누수 방지

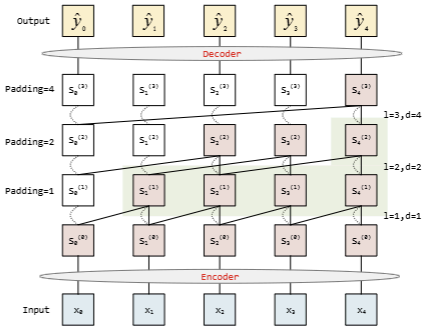
1. Overview

- TCN 구조: Causal Conv + Dilated Conv + Residual Block
- 입력/출력 길이 동일, 시퀀스 전체를 한 번에 처리 가능
- Residual 연결 → 깊은 네트워크 학습 안정화
- Receptive field 조절 → kernel size, dilation 조합으로 long-term dependency 확보

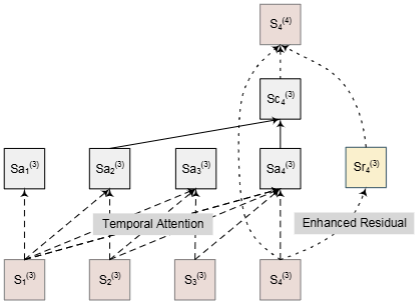
2. Challenges

- RNN 계열: 순차 처리 때문에 학습 느리고 gradient 문제 발생
- Long-range dependency → RNN/LSTM은 멀리 있는 시퀀스 정보 반영 어려움
- Training stability → 깊은 RNN에서는 gradient vanishing/exploding 문제
- TCN: convolution + residual 구조로 안정적 학습, long-term 정보 처리 가능

3. Method



(a) Model overall



(b) TCAN block

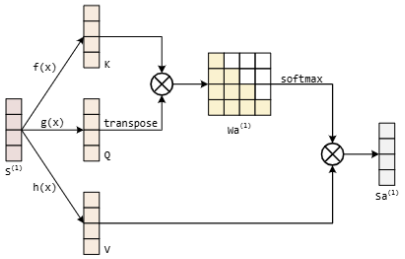


Figure 2: Temporal Attention Block

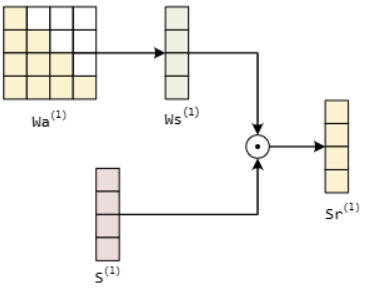


Figure 3: Enhanced Residual Block

- Causal Convolution → 출력 시점 t에서 미래 입력 x(t+1) 이상 사용하지 않음
- Dilated Convolution → dilation factor d 적용, 멀리 있는 시퀀스 정보 반영
  - 수식: y(t)=∑i=0k−1f(i)·x(t−d·i)y(t) = \sum_{i=0}^{k-1} f(i) \cdot x(t - d \cdot i)y(t)=∑i=0k−1f(i)·x(t−d·i)
- Residual Block → 입력과 출력 합산, gradient 흐름 개선
- Network depth + dilation schedule → receptive field 길이 결정
- Hyperparameter: kernel size, layer 수, dilation schedule 중요

4. Experiments

- Synthetic tasks: adding, copy memory → long-term dependency 테스트

- Real-world tasks: PTB char-level, sequential MNIST, music generation, human action prediction
- 비교 모델: LSTM, GRU, vanilla RNN
- 동일 optimizer/lr로 공평 비교
- TCN: receptive field 늘려 long-term 정보 캡처, kernel/dilation 조절 가능

5. Results

Word-level Penn Treebank (PTB)		
Models	Size	ppl^l
Generic TCN [Bai *et al.*, 2018]	13M	88.68
NAS Cell [Zoph and Le, 2017]	54M	62.4
AWD-LSTM [Merity *et al.*, 2018]	24M	58.8
TrellisNet [Bai *et al.*, 2019]	33M	56.80
TrellisNet-MoS [Bai *et al.*, 2019]	34M	54.19
GPT-2 [Radford *et al.*, 2019]	1542M	35.76
TCAN-no-res	13M	**32.19**
TCAN	13M	**30.28**

Character-level Penn Treebank (PTB)		
Models	Size	ppl^l
Generic TCN [Bai *et al.*, 2018]	3.0M	1.31
IndRNN [Li *et al.*, 2018]	12.0M	1.23
NAS Cell [Zoph and Le, 2017]	16.3M	1.214
AWD-LSTM [Merity *et al.*, 2018]	13.8M	1.175
TrellisNet-MoS [Bai *et al.*, 2019]	13.4M	1.158
TCAN-no-res	4.3M	**1.104**
TCAN	4.3M	**1.092**

WikiText-2 (WT2)		
Models	Size	ppl^l
Generic TCN [Bai *et al.*, 2018]	28.6M	138.5
AWD-LSTM [Merity *et al.*, 2018]†	33M	44.3
AWD-LSTM-MoS [Yang *et al.*, 2018]†	35M	40.68
GPT-2 [Radford *et al.*, 2019]	1542M	18.34
TCAN-no-res	33M	**10.92**
TCAN	33M	**9.20**

ER	TA	L_b	L	Size	ppl^l
✗	✓	1	4	13.2M	36.85
✗	✗	2	4	14.7M	151.98

- Synthetic task: TCN 정확도 높고 학습 빠름 → long-range dependency 효과적
- PTB char / MNIST → RNN 대비 높은 accuracy, 안정적 학습
- Music / human action → 긴 시퀀스 예측에서 성능 우위
- 학습 속도 → convolution 병렬 처리 덕분 빠름
- Hyperparameter 영향: 깊이와 kernel size 커지면 성능 ↑, 계산량 ↑

6. Insight

- Long-term dependency 처리 잘함
- RNN보다 병렬 처리 가능 → 속도 빠름
- Residual 구조로 gradient 안정적

- 구조 단순 → 구현 편함
- Dilation schedule 필요
- 깊이/커널 증가 → 연산량 증가