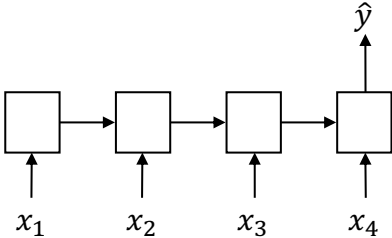
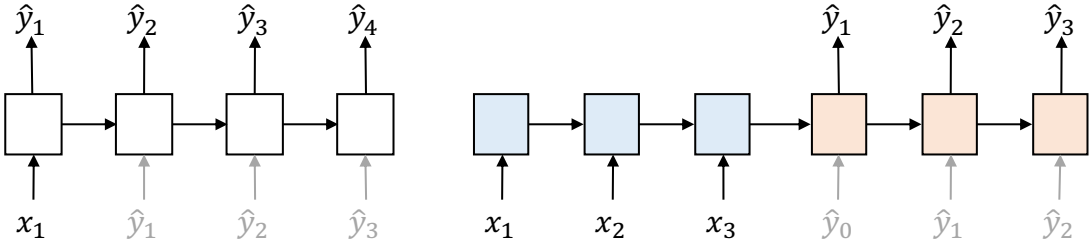
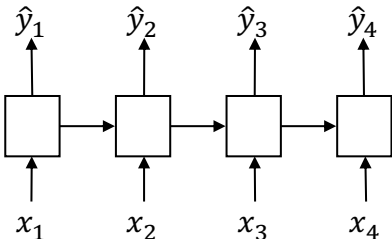


# Auto-regressive & Teacher Forcing

Ki Hyun Kim

[nlp.with.deep.learning@gmail.com](mailto:nlp.with.deep.learning@gmail.com)

# Applications

Type	Architecture	Applications
Many to One		Text Classification
One to Many		NLG, Machine Translation
Many to Many		POS Tagging, MRC

# Two Approaches

## ① Non-autoregressive (Non-generative)

- 현재 상태가 앞/뒤 상태를 통해 정해지는 경우
  - e.g. Part of Speech (POS) Tagging, Text Classification
- Bidirectional RNN 사용 권장

## ① Autoregressive (Generative)

- 현재 상태가 과거 상태에 의존하여 정해지는 경우
  - e.g. Natural Language Generation, Machine Translation
- One-to-Many case 해당
- Bidirectional RNN 사용 불가!!!!

# Auto-regressive

- Inference

$$\hat{x}_t = \operatorname{argmax}_{x_t \in \mathcal{X}} \log P(x_t | \hat{x}_{<t}; \theta)$$

- Auto-regressive:

- 과거 자신의 상태를 참조하여 현재 자신의 상태를 업데이트.

$$\hat{x}_{t=1} = \operatorname{argmax}_{x_t \in \mathcal{X}} \log P(x_{t=1} | x_0; \theta) \text{ where } x_0 = \langle \text{BOS} \rangle.$$

$$\hat{x}_{t=2} = \operatorname{argmax}_{x_t \in \mathcal{X}} \log P(x_{t=2} | x_0, \hat{x}_1; \theta)$$

$$\hat{x}_{t=3} = \operatorname{argmax}_{x_t \in \mathcal{X}} \log P(x_{t=3} | x_0, \hat{x}_1, \hat{x}_2; \theta)$$

...

$$\hat{x}_t = \operatorname{argmax}_{x_t \in \mathcal{X}} \log P(x_t | x_0, \hat{x}_{<t}; \theta)$$

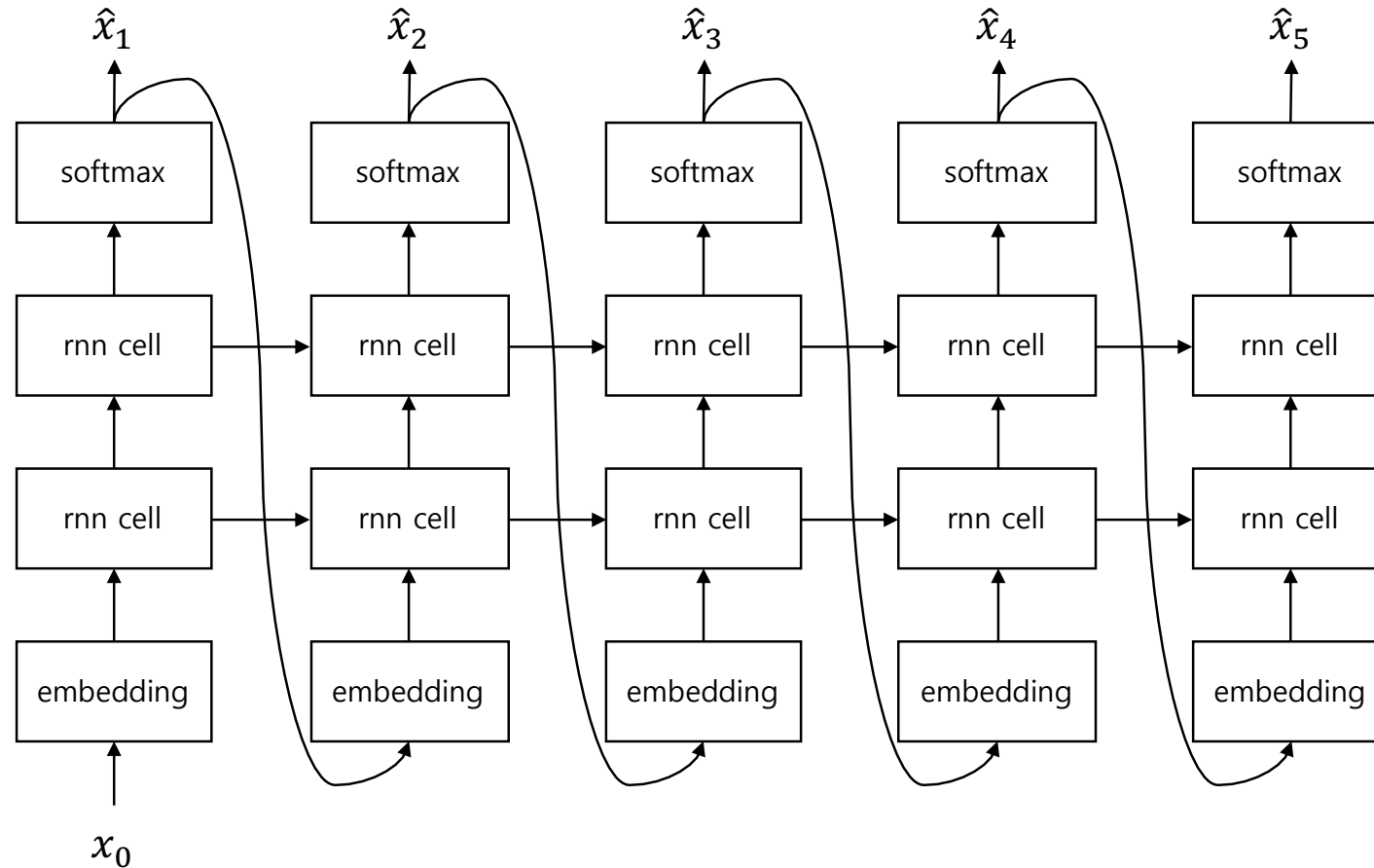
# Teacher-Forcing

- MLE의 수식상, 정답  $x_{t-1}$  을 RNN의 입력으로 넣어줘야 함

$$\begin{aligned}\mathcal{D} &= \{x^i\}_{i=1}^N \\ \hat{\theta} &= \operatorname{argmax}_{\theta \in \Theta} \sum_{i=1}^N \log P(x^i; \theta) \\ &= \operatorname{argmax}_{\theta \in \Theta} \sum_{i=1}^N \sum_{j=1}^n \log P(x_j^i | x_{<j}^i; \theta), \\ \text{where } x^i &= x_{1:n}^i = \{x_1^i, \dots, x_n^i\}.\end{aligned}$$

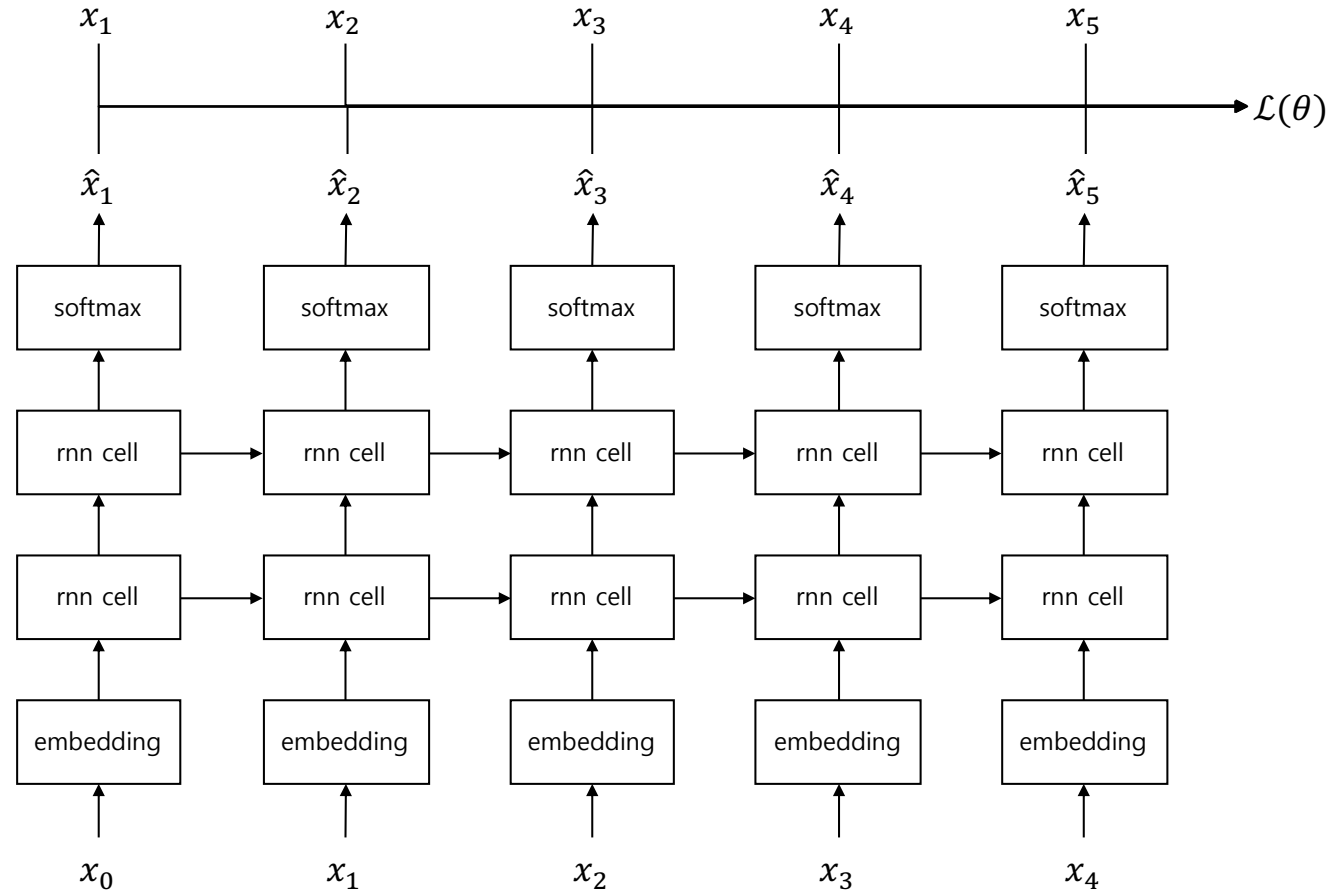
# Auto-regressive and Teacher Forcing

- Inference Mode



# Teacher Forcing

- Training Mode



# 고통의 시작: NLG is Auto-regressive Task

- Auto-regressive task에서는 보통 이전 time-step의 모델을 출력을 다음 time-step의 입력으로 넣어 줌
  - 이전 time-step의 출력에 따라 현재 모델의 state가 바뀌게 될 것
- 하지만 적절한 학습을 위해서는 학습 시에는 이전 time-step의 출력 값이 아닌, 실제 정답을 넣어 줌
- 따라서 학습과 추론을 위한 방법이 다르게 되어 여러가지 문제가 발생
  - 학습을 위한 코드와 추론을 위한 코드를 따로 짜야 함
  - 학습과 추론 방법의 괴리(discrepancy)가 발생하여 성능이 저하될 수 있음