

Lec. 14 Recurrent Neural Networks

Keywords: Memory, State, BPTT

Will Cover

1. Introduction

- Types of tasks processing 'sequences'
- Neural Language Model to RNN

2. Recurrent Neural Nets

- Unrolling network to understand like FFNN
- 3 examples of how parameter setting results in RNN

3. Backprop through time

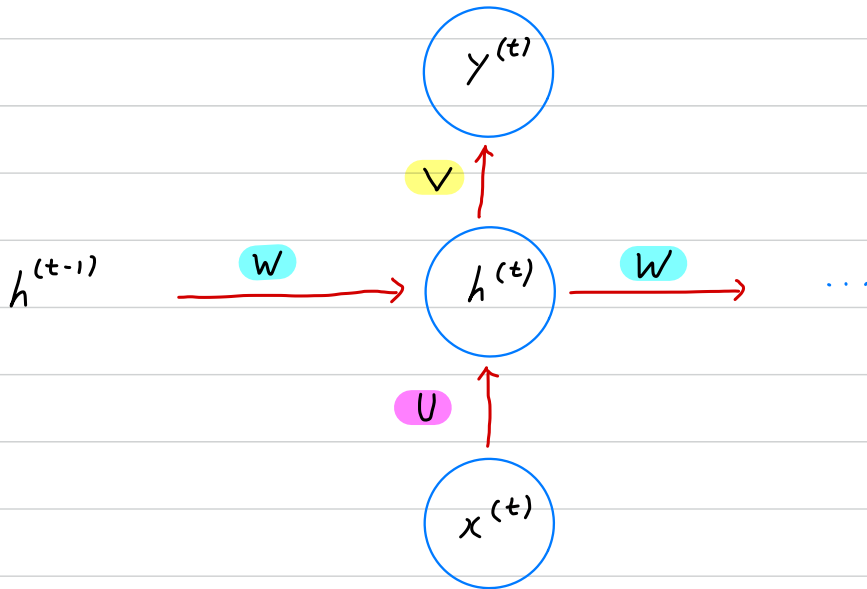
- View as MLP backprop with unrolled computation graph
- Comparing with MLP backprop

4. Sequence Modeling (what task can RNN be applied)

- Language Modeling
- Neural Machine Translation
- Learning to execute Programs

3. Back prop through time

Before BPTT, check forward pass



1. sum

$$z^{(t)} = h^{(t-1)} \cdot W + x^{(t)} \cdot U$$

2. activation

$$h^{(t)} = \phi_1(z^{(t)})$$

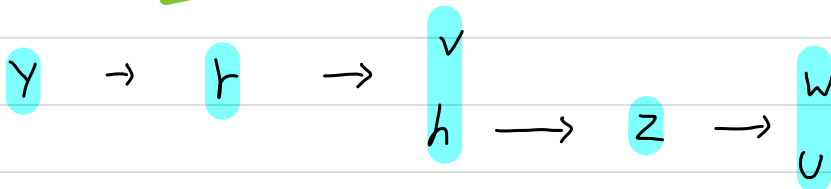
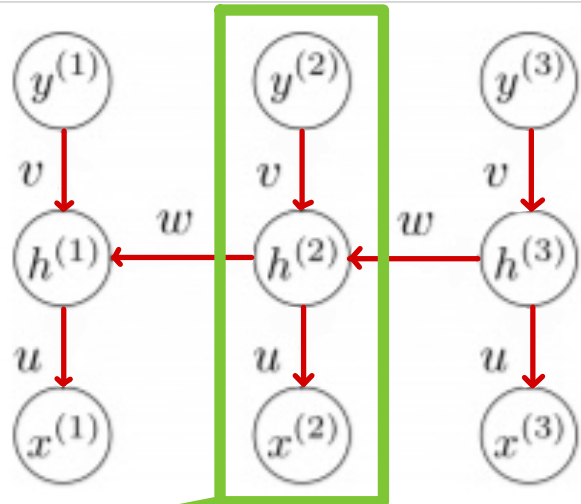
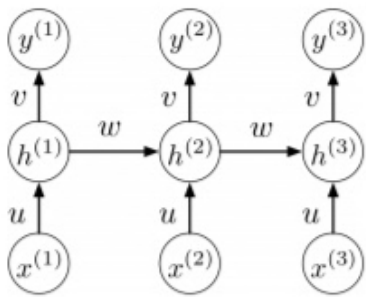
3. weighting

$$r^{(t)} = V h^{(t)}$$

4. activating

$$y^{(t)} = \phi_2(r^{(t)})$$

Back prop \rightarrow switch arrow direction !



for ward

Back ward

$$L = \sum_m^3 \gamma^{(m)}$$

BP \rightarrow

$$\overline{\gamma^{(t)}} = \frac{1}{m} \cdot L$$

$$\gamma^{(t)} = \phi_2(r^{(t)})$$

BP \rightarrow

$$\overline{r^{(t)}} = \overline{\gamma^{(t)}} \cdot \phi_2'(r^{(t)})$$

$$r^{(t)} = v \cdot h^{(t)}$$

BP \rightarrow

$$\overline{v} = \overline{r^{(t)}} \cdot h^{(t)}$$

$$r^{(t)} = v \cdot h^{(t)}$$

BP \rightarrow

$$\overline{h^{(t)}} = v \cdot \overline{r^{(t)}} +$$

$$z^{(t+1)} = w \cdot h^{(t)} + u \cdot x^{(t+1)}$$

$$w \cdot \overline{z^{(t+1)}}$$

$$h^{(t)} = \phi_1(z^{(t)})$$

BP \rightarrow

$$\overline{z^{(t)}} = \overline{h^{(t)}} \cdot \phi_1'(z^{(t)})$$

$$z^{(t)} = w \cdot h^{(t-1)} + u \cdot x^{(t)}$$

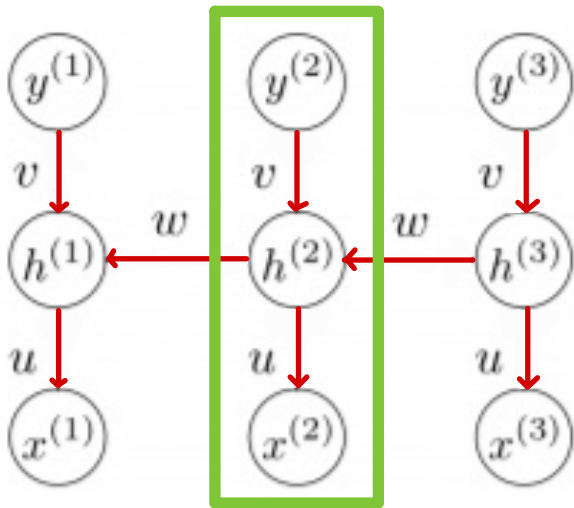
BP \rightarrow

$$\overline{w} = \overline{z^{(t)}} \cdot h^{(t-1)}$$

$$z^{(t)} = w \cdot h^{(t-1)} + u \cdot x^{(t)}$$

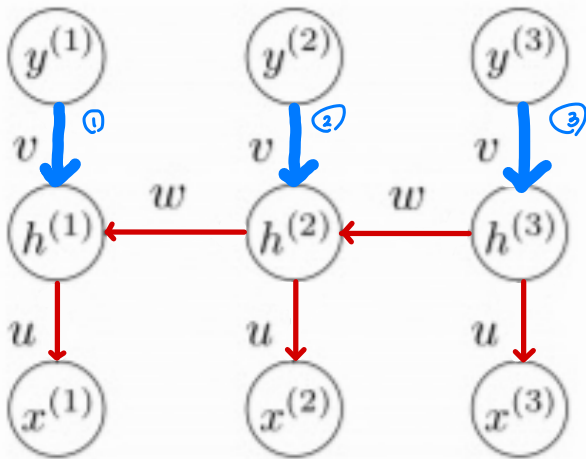
BP \rightarrow

$$\overline{u} = \overline{z^{(t)}} \cdot x^{(t)}$$



높은 구간이 아닌 전 구간으로
효과 상충하여 생각해 보죠 ?

V

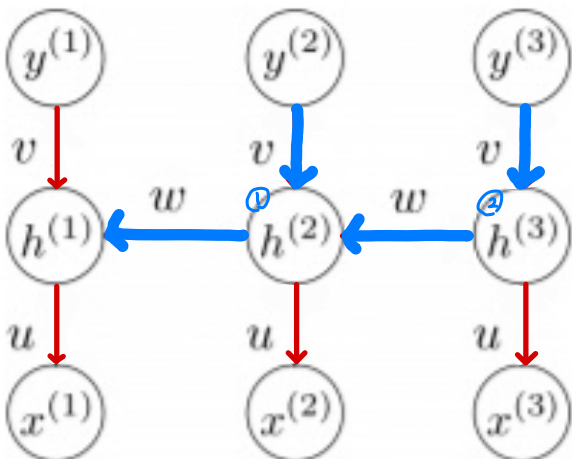


$$\bar{V} = \bar{1} + \bar{2} + \bar{3}$$

Generally,

$$\bar{V} = \sum_{t=1}^T \bar{r}^{(t)} \cdot h^{(t)}$$

W

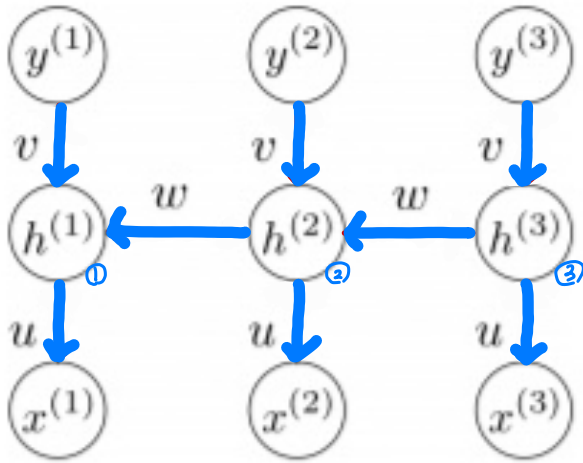


$$\bar{W} = \bar{1} + \bar{2}$$

Generally,

$$\bar{W} = \sum_{t=1}^{T-1} \bar{z}^{(t+1)} \cdot h^{(t)}$$

U



$$\overline{U} = \overline{①} + \overline{②} + \overline{③}$$

Generally,

$$\overline{U} = \sum_{t=1}^T \overline{z^{(t)}} \cdot x^{(t)}$$

Vectorized representation ($N = \text{batch size}$)

$$\overline{U} = \frac{1}{N} \sum_{t=1}^T \overline{Z^{(t)}}^T \mathbf{X}^{(t)}$$

$$\overline{V} = \frac{1}{N} \sum_{t=1}^T \overline{R^{(t)}}^T \mathbf{H}^{(t)}$$

$$\overline{W} = \frac{1}{N} \sum_{t=1}^{T-1} \overline{Z^{(t+1)}}^T \mathbf{H}^{(t)}$$

\Rightarrow MLP Back prop 과 계산 과정은 동일하다.

Unrolled-graph 에서 직전 노드에 전달도!

Gradient 와 두 변수 사이 미분값이 곱해져 다음 노드에

전달된다. 다만 Unrolled graph 에서 각 가중치는 개별 변수가

아니라, 동일 가중치 이므로 각 시점에서의 gradient 가 sum 되는

차이만 있을 뿐.

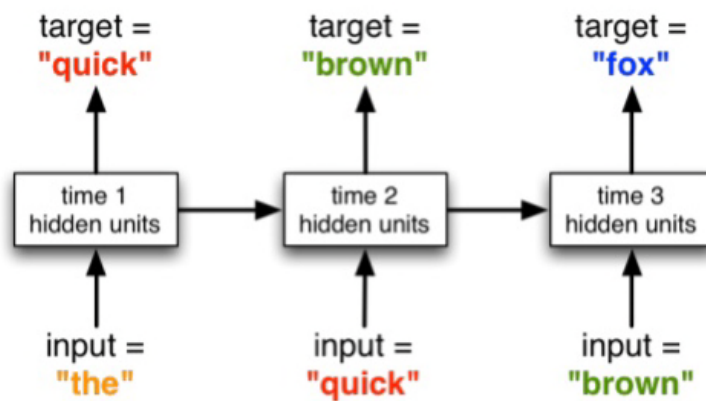
4. Sequence Modeling (What task can RNN be applied)

4.1 Language Modeling

RNN 을 통해 LM 을 진행할 수 있다.

N -gram 모델, Neural Language 모델과 동일하게 입력 시퀀스에
확률을 할당한다.

그러나 이전 모델과는 달리 이전 N 개의 정보에 의해서만 현재 정보에
확률 할당이 가능할 것이라는 마르코브 가정을 하지 않고,
현재 시점의 정보는 이전 모든 정보의 영향을 받아 확률이 할당된다고
가정한다.



생성 모델로서 RNN 을 효율하게 되면, 이전 $timestep$ 의
출력을 입력으로 활용하는 방식으로 적용 가능하다.



그리고 RNN 을 통한 Language modeling 에는 한가지

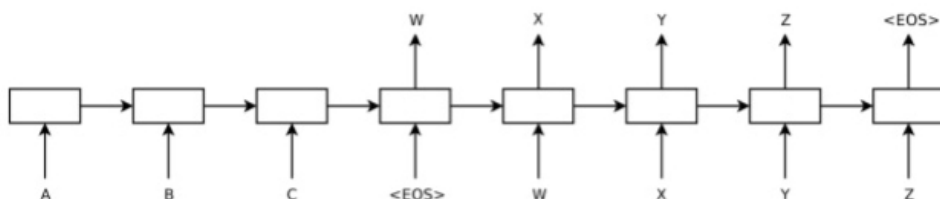
더 이점이 존재하는데, 바로 character-level modeling 이 가능하다는 것이다. 한정도! output set 을 통해 메모리 이득을 취할 수 있다.

ex)

He was elected President during the Revolutionary War and forgave Opus Paul at Rome. The regime of his crew of England, is now Arab women's icons in and the demons that use something between the characters' sisters in lower coil trains were always operated on the line of the **ephemeral** street, respectively, the graphic or other facility for deformation of a given proportion of large segments at RTUS). The B every chord was a "strongly cold internal palette pour even the white blade."

4.2 Neural Machine Translation

기존 machine translation task 는 쌍으로 이루어진 문장 쌍에 대해 직접 변환하려는 알고리즘에 의해 이루어 졌으나 , RNN 의 등장으로 다른 형태의 seq-to-seq task 접근이 가능해짐.



바로 입력 테이터를 semantic space 에 mapping 하는 네트워크와 semantic vector 로 부터 특정 도메인에서 generate 하는 네트워크를 엮는 것 !!