Neural Language Models

SLP 7.5

https://web.stanford.edu/~jurafsky/slp3/7.pdf

Natural Language Processing in Action Ch. 7, 8, 9

Outline

- Neural networks for **language modeling** -- assigning probabilities to word sequences and predicting upcoming words

- Sequential models, e.g., Recurrent Neural Network, LSTM

Neural Language Models

- Use a neural network as a probabilistic classifier, to **compute the probability of the next word** **given the previous _n_ words**

$$P(w_5|w_1,w_2,w_3,w_4)$$
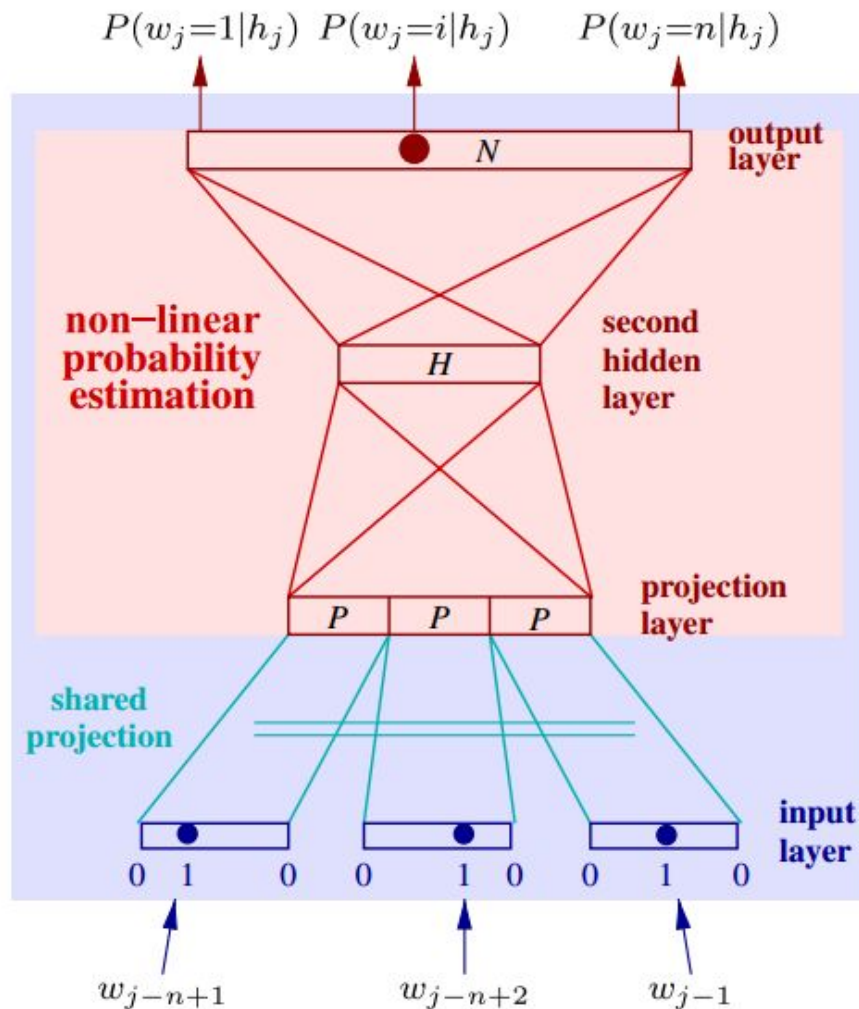
Feedforward Neural LMs (Bengio et al., 2003)

- Input at time t: a representation of previous words (w_{t-1}, w_{t-2}, etc)
- Output: a probability distribution over possible next words
- Feedforward neural LM approximates the probability of a word given the entire prior context $P(w_t \mid w_1^{t-1})$ **by approximating based on the N previous words**

$$P(w_t | w_1^{t-1}) \approx P(w_t | w_{t-N+1}^{t-1})$$

Use a neural network as a probabilistic classifier, to **compute the probability of the next word** given the **previous *n* words**
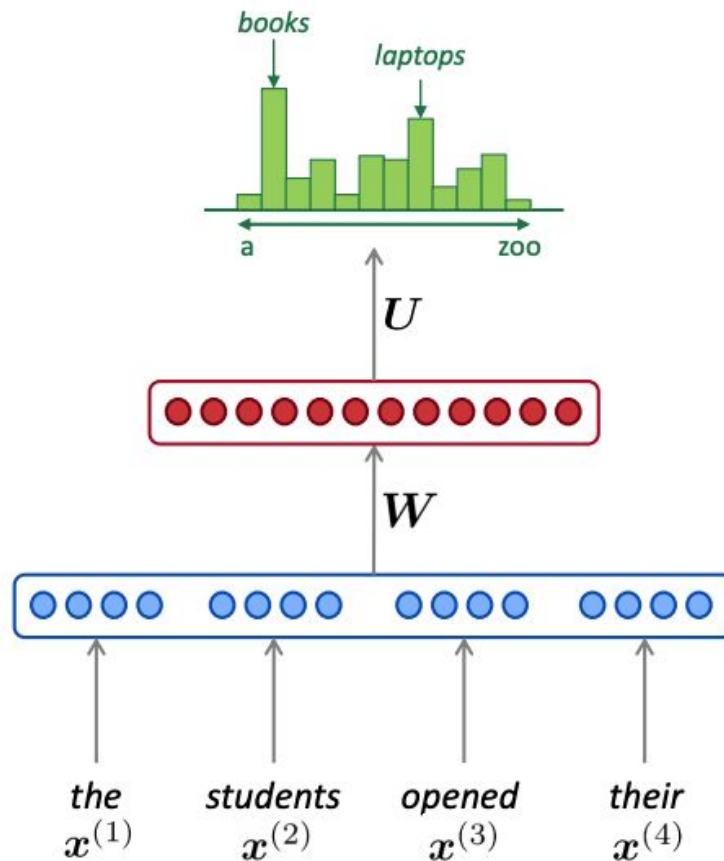
$P(w_t = i \mid w_{t-1}, w_{t-2}, w_{t-3}, w_{t-4})$

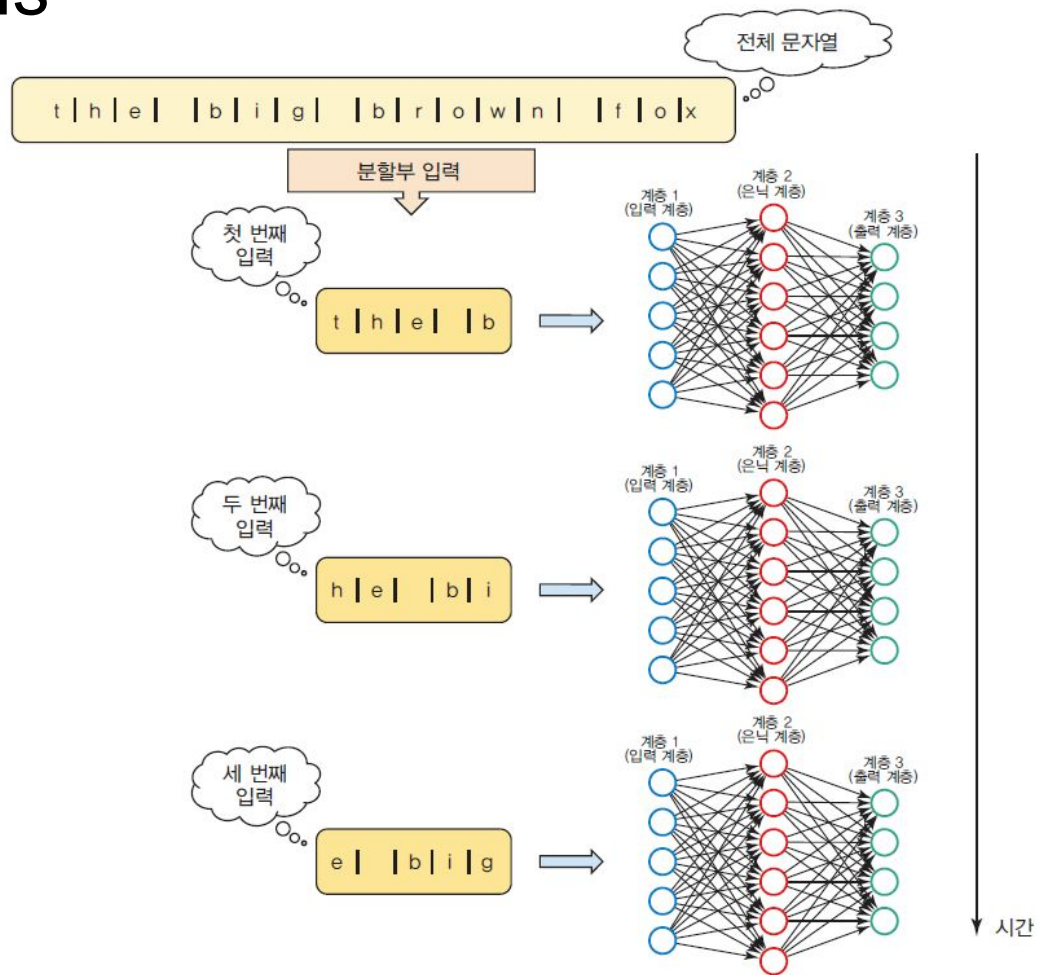$P(w_j=1|h_j)$ $P(w_j=i|h_j)$ $P(w_j=n|h_j)$

Neural Network

non−linear probability estimation

shared projection

output layer

N

second hidden layer

H

projection layer

P P P

input layer

0 1 0 0 1 0 0 1 0

w_{j-n+1} w_{j-n+2} w_{j-1}

Feedforward Neural LMs

Word-based Feedforward LM

Feedforward Neural LMs

Character-based
Feedforward LM

Sequence Processing with Recurrent Networks

SLP Ch. 9
https://web.stanford.edu/~jurafsky/slp3/9.pdf
NLP in Action Ch. 8

Limitations of Feedforward Neural LM

- The models accept a small **fixed-sized window of tokens as input**

- **Limit the context from which information can be extracted**; anything outside the context window has no impact on the decision being made

- Recurrent neural networks process sequences explicitly as sequences, to handle variable length inputs without the use of fixed-sized windows
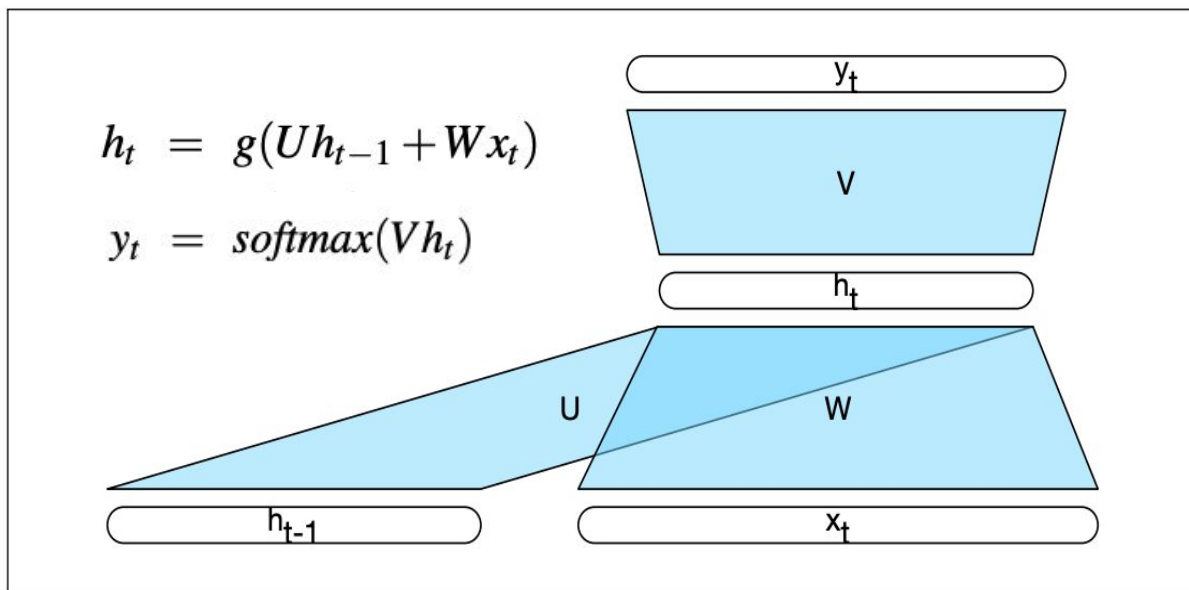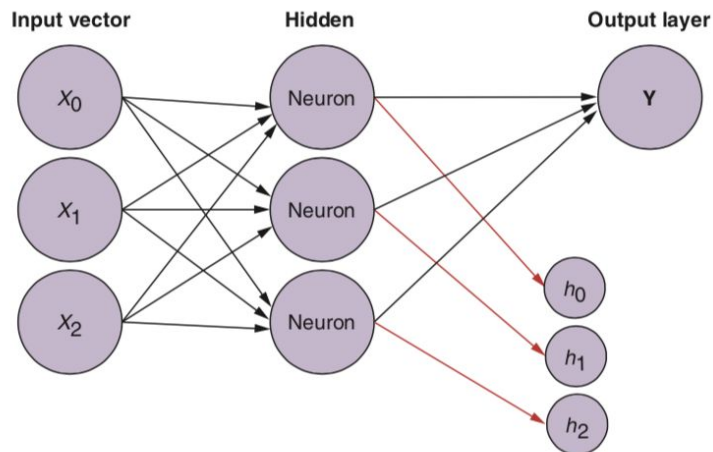
Simple Recurrent Networks (SRN)
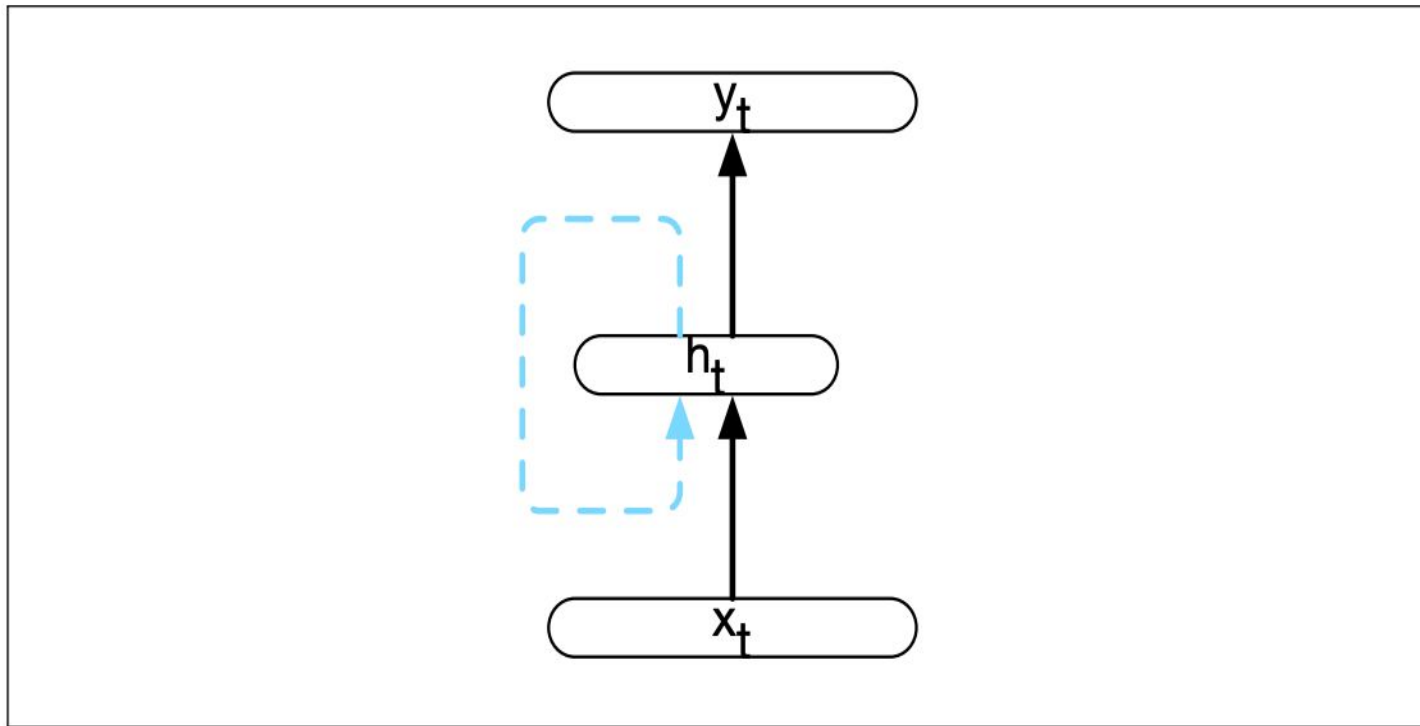
Simple Recurrent Networks (SRN)

Learn U, V, W

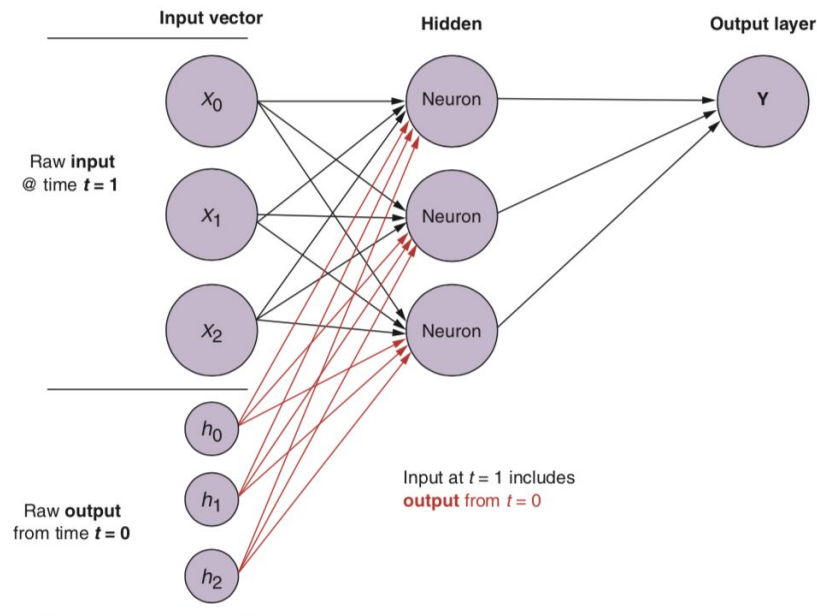$$h_t = g(Uh_{t-1} + Wx_t)$$

$$y_t = softmax(Vh_t)$$
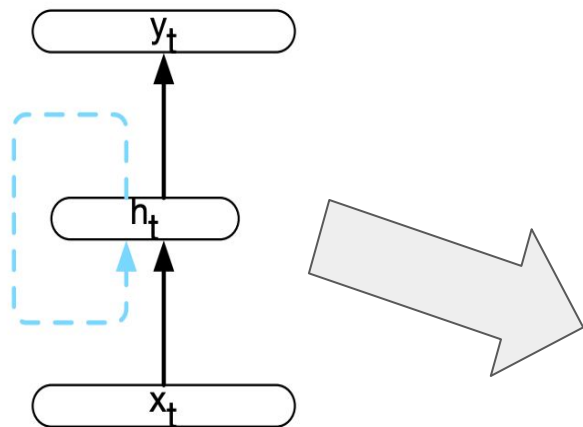
RNN Language Model

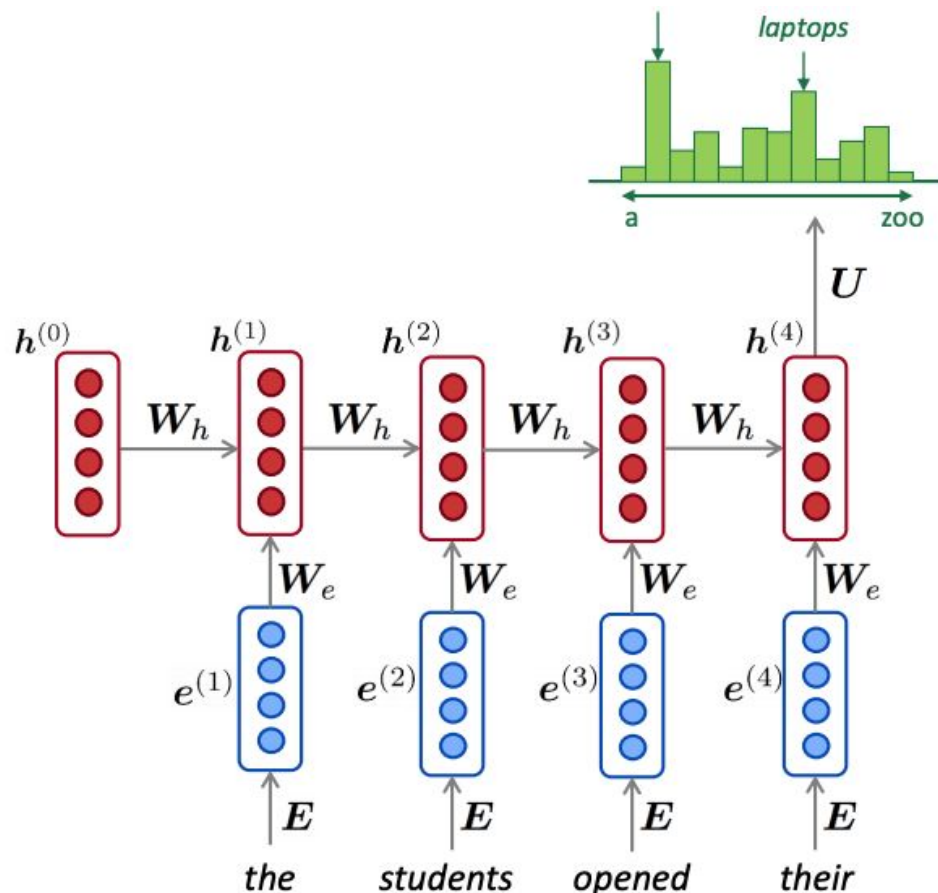At time step 0, input the initial token alongside a 0 vector

RNN at time step t = 0

RNN at time step t = 1

RNN Language Model

Word-based RNN LM

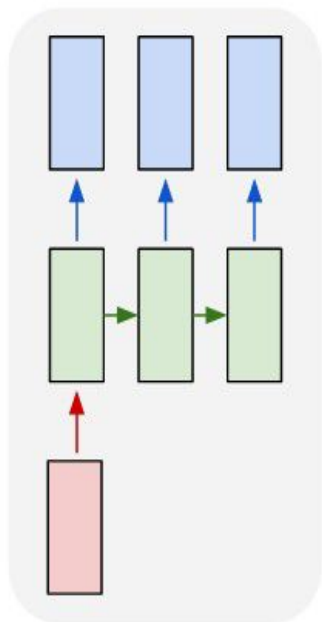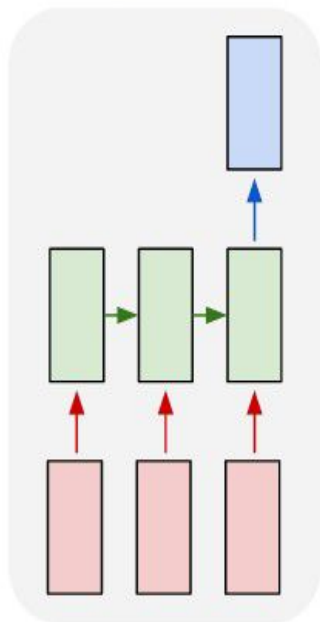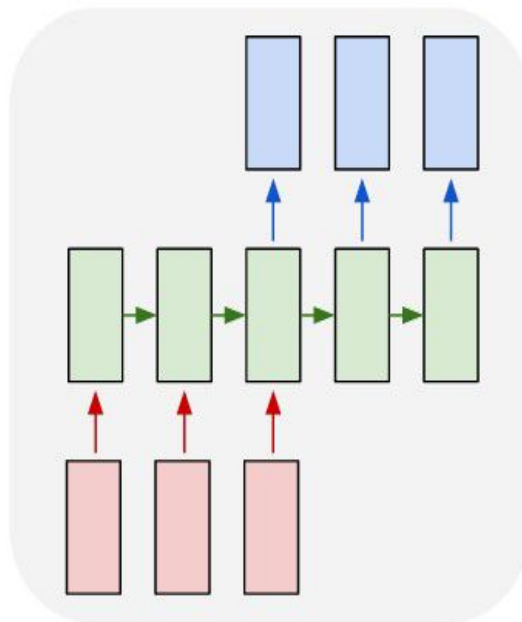# Different RNN sequence types

one to many   many to one   many to many   many to many