natural language processing (NLP) and recurrent neural networks(RNN)

stats403_deep_learning

spring_2025

lecture_5

5.1 background from NLP

how to represent texts?

John likes to watch movies. Mary likes movies too.

how to represent texts?

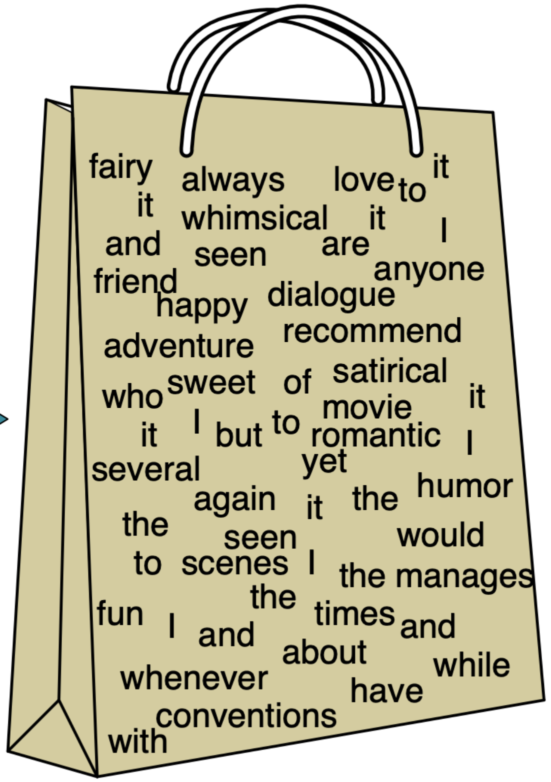
John likes to watch movies. Mary likes movies too.

{"John":1,"likes":2,"to":1,"watch":1,"movies":2,
"Mary":1,"too":1}

how to represent texts?

- bag-of-words

I love this movie! It's sweet, but with satirical humor. The dialogue is great and the adventure scenes are fun... It manages to be whimsical and romantic while laughing at the conventions of the fairy tale genre. I would recommend it to just about anyone. I've seen it several times, and I'm always happy to see it again whenever I have a friend who hasn't seen it yet!



it	6
I	5
the	4
to	3
and	3
seen	2
yet	1
would	1
whimsical	1
times	1
sweet	1
satirical	1
adventure	1
genre	1
fairy	1
humor	1
have	1
great	1
…	..

tokenization

- tokenization: breaking down a sequence of text into individual units called tokens

- in English, words are mostly segmented by spaces and punctuation.
  - exceptions: New York, rock 'n' roll
- Penn Treebank tokenization standard:

Input: `"The San Francisco-based restaurant," they said,`
`"doesn’t charge $10".`

Output: `"␣The␣San␣Francisco-based␣restaurant␣,␣"␣they␣said␣,␣`
`"␣does␣n’t␣charge␣$␣10␣"␣.`

tokenization

- Hanzi (Chinese characters):
  - what counts as a word in Chinese is complex

姚明进入总决赛
"Yao Ming reaches the finals"

姚明 进入 总决赛
YaoMing reaches finals

"Chinese Treebank" segmentation

姚 明 进入 总 决赛
Yao Ming reaches overall finals

"Peking University" segmentation

a reasonable semantic level for most applications

姚 明 进 入 总 决 赛
Yao Ming enter enter overall decision game

tokenization

- byte pair encoding (BPE)

    1. start with individual characters (bytes) as the base vocabulary
    2. count frequent pairs of adjacent symbols
    3. merge the most frequent pair into a new symbol
    4. repeat steps 2–3 until you reach the desired vocabulary size

lemmatization and stemming

- lemmatization (词形还原) and stemming (词干提取) are techniques used in natural language processing to reduce words to their base or root forms, making it easier to analyze and compare text data.

- lemmatization is the process of reducing words to their base or dictionary form, known as the “lemma”. The lemma is a valid word that represents the original word.
- stemming is the process of removing prefixes or suffixes from words to obtain the word's root form, or the “stem”.

lemmatization and stemming

- The boy's cars are different colors

- The boy car be differ color

This was not the map we found in Billy Bones's chest, but an accurate copy, complete in all things-names and heights and soundings-with the single exception of the red crosses and the written notes.

Thi wa not the map we found in Billi Bone s chest but an accur copi complet in all thing name and height and sound with the singl except of the red cross and the written note

how to represent words?

荃者所以在魚，得魚而忘荃
Nets are for fish; once you get the fish, you can forget the net.

言者所以在意，得意而忘言
Words are for meaning; once you get the meaning, you can forget the words.

(莊子Zhuangzi: Chapter 26)

quoted in Jurafsky, D., & Martin, J. H. (2018). *Speech and language processing.*

how to represent words?

- tezgüino

how to represent words?

A bottle of ____ is on the table.

Everybody likes ____.

Don't have ____ before you drive.

We make ____ out of corn.

how to represent words?

A bottle of ____ is on the table.

Everybody likes ____.

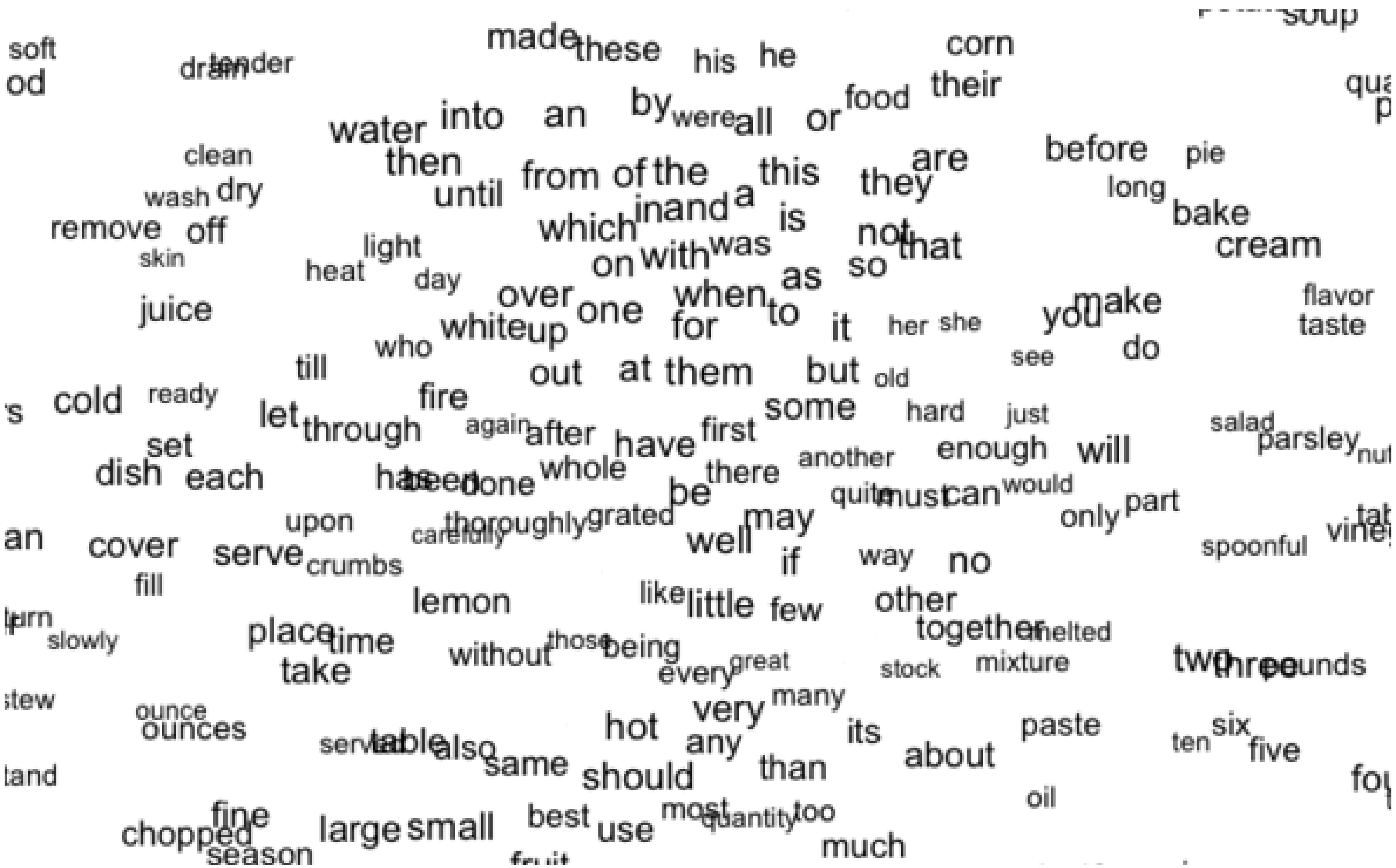
Don't have ____ before you drive.

We make ____ out of corn.

tezgüino	1	1	1	1
loud	0	0	0	0
motor oil	1	0	0	1
tortillas	0	1	0	1
choices	0	1	0	0
wine	1	1	1	0

how to represent words?

- “words” as “vectors”
- similar words should be “neighbors”



https://www.adityathakker.com/introduction-to-word2vec-how-it-works/

how to represent words?

- “words” as “vectors”

- naïve idea: “one-hot” vectors
  - does not contain information about “similarity”

		cat	mat	on	sat	the
the	=>	0	0	0	0	1
cat	=>	1	0	0	0	0
sat	=>	0	0	0	1	0
...				...		

https://www.tensorflow.org/text/guide/word_embeddings

how to represent words?

- co-occurrence matrix
    - ❖frequency is not the best measure of association between words
    - ❖very high dimension

- I enjoy flying.
- I like NLP.
- I like deep learning.

counts	I	like	enjoy	deep	learning	NLP	flying	.
I	0	2	1	0	0	0	0	0
like	2	0	0	1	0	1	0	0
enjoy	1	0	0	0	0	0	1	0
deep	0	1	0	0	1	0	0	0
learning	0	0	0	1	0	0	0	1
NLP	0	1	0	0	0	0	0	1
flying	0	0	1	0	0	0	0	1
.	0	0	0	0	1	1	1	0

tf-idf

- term frequency – inverted document frequency
- this index measures how important a word is to a document in a collection or corpus

number of times 'term' appears in 'document'

$$\mathrm{tf}(t,d) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}}$$

total number of terms in 'document'

total number of documents

$$\mathrm{idf}(t,D) = \log \frac{N}{|\{d \in D : t \in d\}|}$$

number of documents containing 'term'

$$\mathrm{tfidf}(t,d,D) = \mathrm{tf}(t,d) \cdot \mathrm{idf}(t,D)$$

tf-idf

	As You Like It	Twelfth Night	Julius Caesar	Henry V
battle	0.074	0	0.22	0.28
good	0	0	0	0
fool	0.019	0.021	0.0036	0.0083
wit	0.049	0.044	0.018	0.022

word2vec

Efficient Estimation of Word Representations in Vector Space

Tomas Mikolov
Google Inc., Mountain View, CA
tmikolov@google.com

Kai Chen
Google Inc., Mountain View, CA
kaichen@google.com

Greg Corrado
Google Inc., Mountain View, CA
gcorrado@google.com

Jeffrey Dean
Google Inc., Mountain View, CA
jeff@google.com

- continuous bag of words model (CBOW)
  - predict the target word (middle word) based on surrounding context words
- continuous skip-gram model
  - predict the surrounding words based on the target word

continuous bag of words model (CBOW)

Window Size	Text
2	[The **wide** road shimmered] in the hot sun.
	The [wide road **shimmered** in the] hot sun.
	The wide road shimmered in [the hot **sun**].
3	[The **wide** road shimmered in] the hot sun.
	[The wide road **shimmered** in the hot] sun.
	The wide road shimmered [in the hot **sun**].

(the, road, shimmered) , wide

(wide, road, in, the) , shimmered

(the, hot) , sun

continuous skip-gram model

Window Size	Text	Skip-grams
2	[The **wide** road shimmered] in the hot sun.	wide, the wide, road wide, shimmered
	The [wide road **shimmered** in the] hot sun.	shimmered, wide shimmered, road shimmered, in shimmered, the
	The wide road shimmered in [the hot **sun**].	sun, the sun, hot
3	[The **wide** road shimmered in] the hot sun.	wide, the wide, road wide, shimmered wide, in
	[The wide road **shimmered** in the hot] sun.	shimmered, the shimmered, wide shimmered, road shimmered, in shimmered, the shimmered, hot
	The wide road shimmered [in the hot **sun**].	sun, in sun, the sun, hot

word2vec

center word lookup matrix

$$V = \begin{array}{c} \begin{bmatrix} | & | & & | & | \\ v_0 & v_1 & \cdots & v_{n-1} & v_n \\ | & | & & | & | \end{bmatrix} \\ \begin{array}{ccccc} \text{wide} & & & \text{road} & \end{array} \end{array}$$

outer word lookup matrix

$$U = \begin{array}{c} \begin{bmatrix} | & | & & | & | \\ u_0 & u_1 & \cdots & u_{n-1} & u_n \\ | & | & & | & | \end{bmatrix} \\ \begin{array}{ccccc} \text{wide} & & & \text{road} & \end{array} \end{array}$$

each column here is embedding of a word

need: use the training text to determine the best V and U, and use e. g. $V + U$ as the word vectors

5.2 (cute) language models

probabilistic modeling

- suppose we want to translate Spanish into English:
    - ❑ El cafe negro me gusta mucho.
    - ❑ The coffee black me pleases much.

probabilistic modeling

- a good language model of English will tell us

$$\mathrm{p}(\textit{The coffee black me pleases much}) < \mathrm{p}(\textit{I love dark coffee})$$

noisy channel model

- language model: $\mathrm{p}_e(\boldsymbol{w}^{(e)})$
- translation model: $\mathrm{p}_{s|e}(\boldsymbol{w}^{(s)} \mid \boldsymbol{w}^{(e)})$

$$
\begin{aligned}
\mathrm{p}_{e|s}(\boldsymbol{w}^{(e)} \mid \boldsymbol{w}^{(s)}) \propto & \mathrm{p}_{e,s}(\boldsymbol{w}^{(e)}, \boldsymbol{w}^{(s)}) \\
= & \mathrm{p}_{s|e}(\boldsymbol{w}^{(s)} \mid \boldsymbol{w}^{(e)}) \times \mathrm{p}_e(\boldsymbol{w}^{(e)})
\end{aligned}
$$

n-gram model

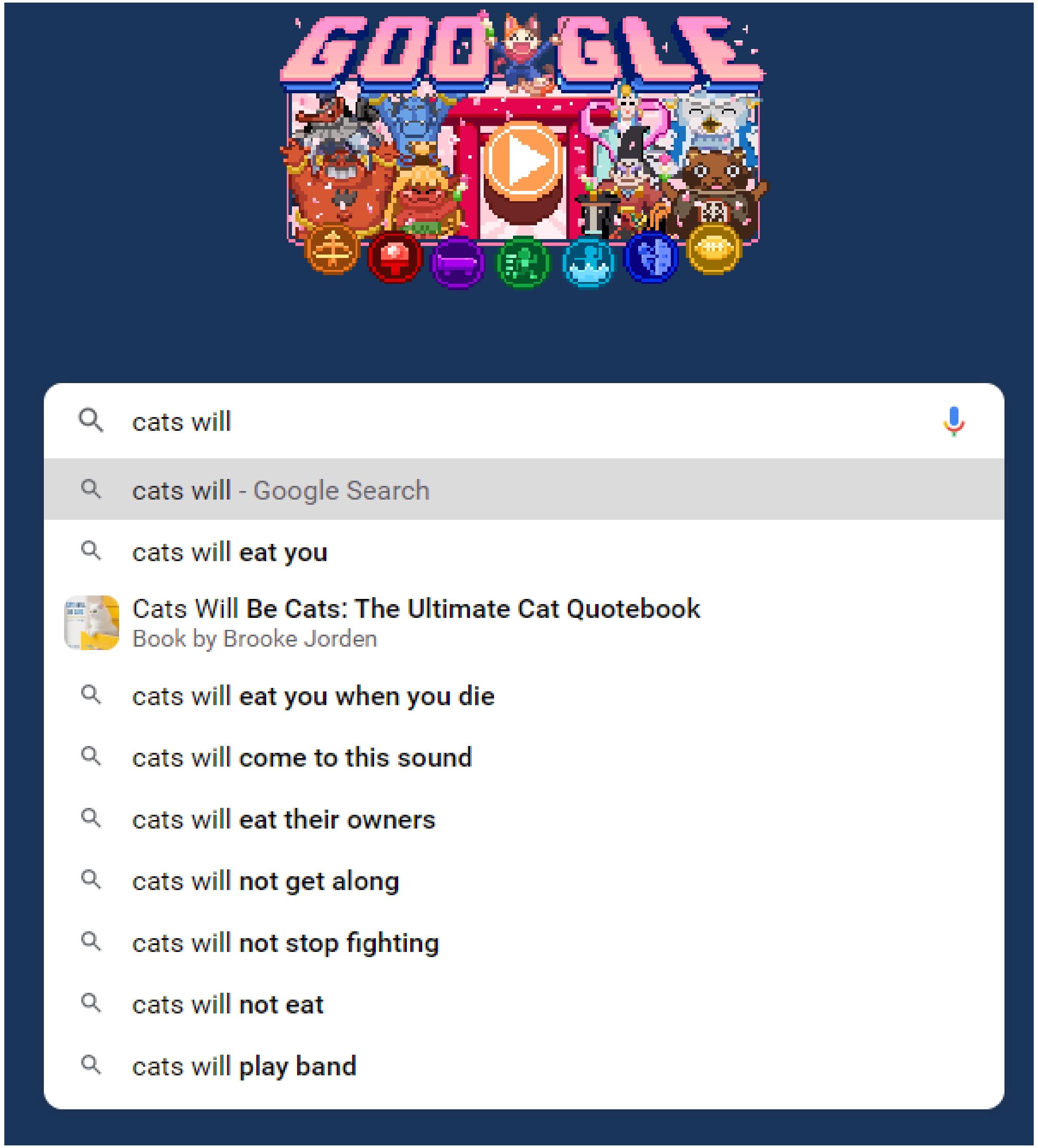
- relative frequency estimate:

$$
\begin{aligned}
&\mathrm{p}(\textit{Computers are useless, they can only give you answers}) \\
&\quad = \frac{\mathrm{count}(\textit{Computers are useless, they can only give you answers})}{\mathrm{count}(\text{all sentences ever spoken})}
\end{aligned}
$$

n-gram model

$$\begin{aligned}
\mathrm{p}(\boldsymbol{w}) =& \mathrm{p}(w_1, w_2, \ldots, w_M) \\
=& \mathrm{p}(w_1) \times \mathrm{p}(w_2 \mid w_1) \times \mathrm{p}(w_3 \mid w_2, w_1) \times \ldots \times \mathrm{p}(w_M \mid w_{M-1}, \ldots, w_1)
\end{aligned}$$

n-gram model


cats will
cats will - Google Search
cats will eat you
Cats Will Be Cats: The Ultimate Cat Quotebook
Book by Brooke Jorden
cats will eat you when you die
cats will come to this sound
cats will eat their owners
cats will not get along
cats will not stop fighting
cats will not eat
cats will play band

n-gram model

- the n-gram model makes the crucial approximation:

$$p(w_m \mid w_{m-1} \dots w_1) \approx p(w_m \mid w_{m-1}, \dots, w_{m-n+1})$$

n-gram model

- a bigram (n=2) approximation would be

$$\mathrm{p}(\textit{I like black coffee}) = \mathrm{p}(\textit{I} \mid \square) \times \mathrm{p}(\textit{like} \mid \textit{I}) \times \mathrm{p}(\textit{black} \mid \textit{like}) \times \mathrm{p}(\textit{coffee} \mid \textit{black}) \times \mathrm{p}(\blacksquare \mid \textit{coffee})$$

5.5 recurrent neural networks (RNN)

feed-forward network for language

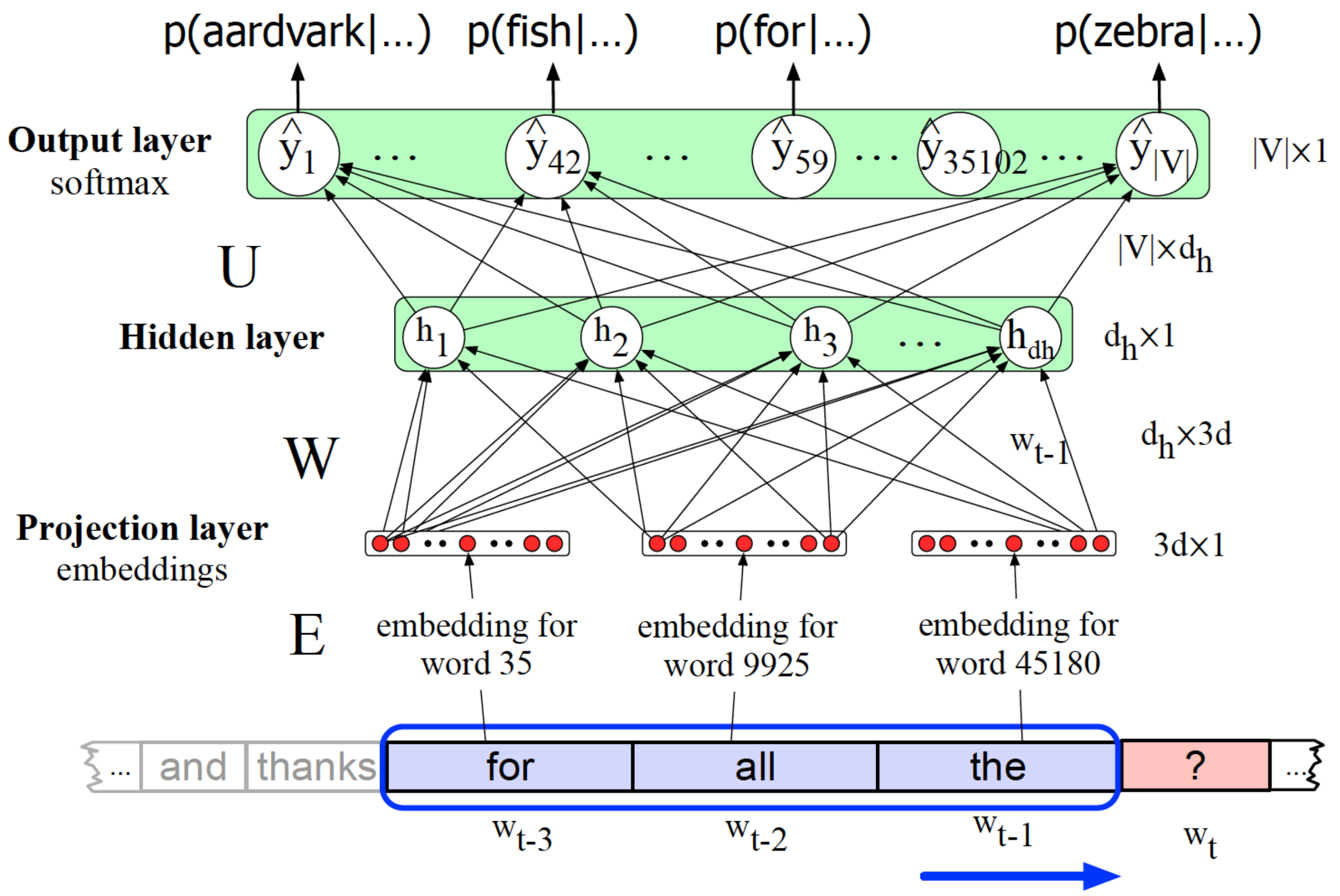


image credit: Jurafsky, D., & Martin, J. H. (2018). *Speech and language processing*.

sequential modeling

- MLP needs to learn many combination following grammar
  - solution?

sequential modeling

- MLP needs to learn many combination following grammar
  - solution: parameter sharing

- 1D CNN?
  - problem?

sequential modeling

- MLP needs to learn many combination following grammar
  - solution: parameter sharing
- 1D CNN?
  - problem: large kernel
- need: treat time sequence naturally

dynamical system

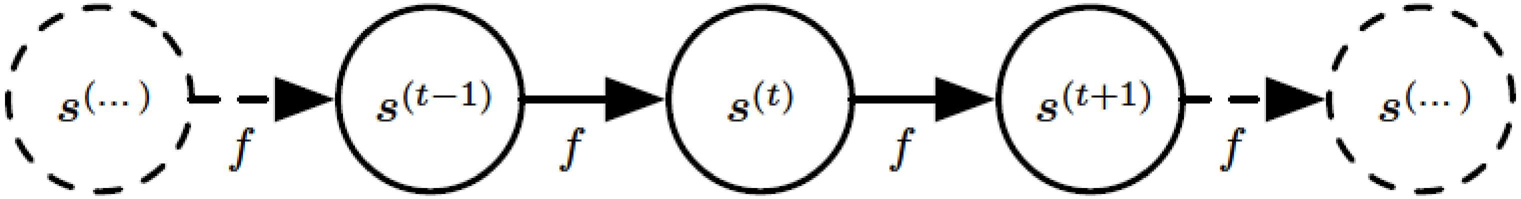
$$\boldsymbol{s}^{(t)} = f(\boldsymbol{s}^{(t-1)}; \boldsymbol{\theta})$$

state of system
at time t

dynamical system

$$\boldsymbol{s}^{(t)} = f(\boldsymbol{s}^{(t-1)}; \boldsymbol{\theta})$$

state of system
at time t

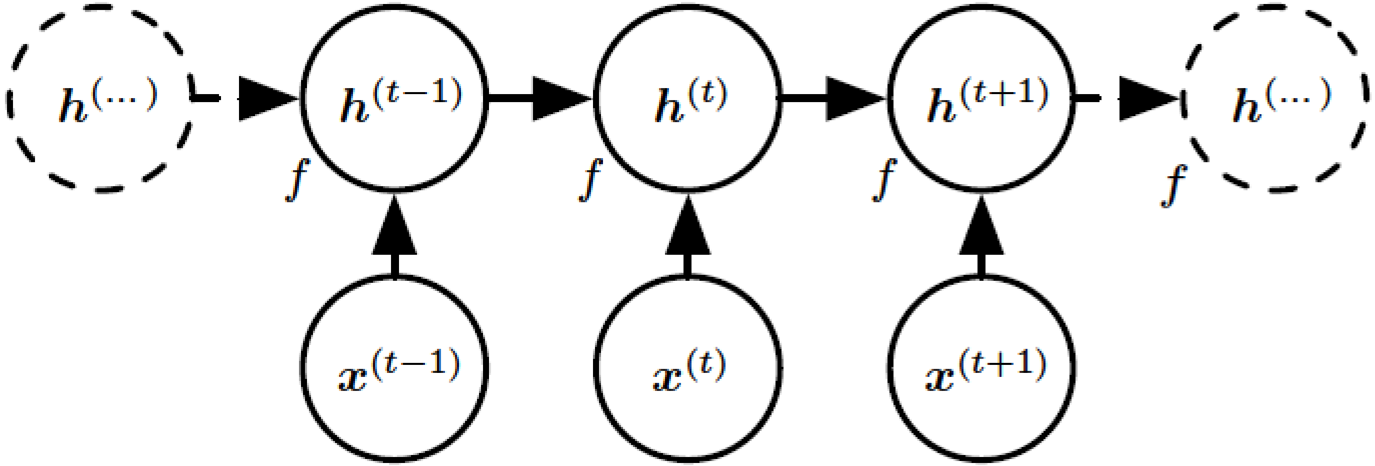


Elman RNN: external input

$$\boldsymbol{h}^{(t)} = f(\boldsymbol{h}^{(t-1)}, \boldsymbol{x}^{(t)}; \boldsymbol{\theta})$$

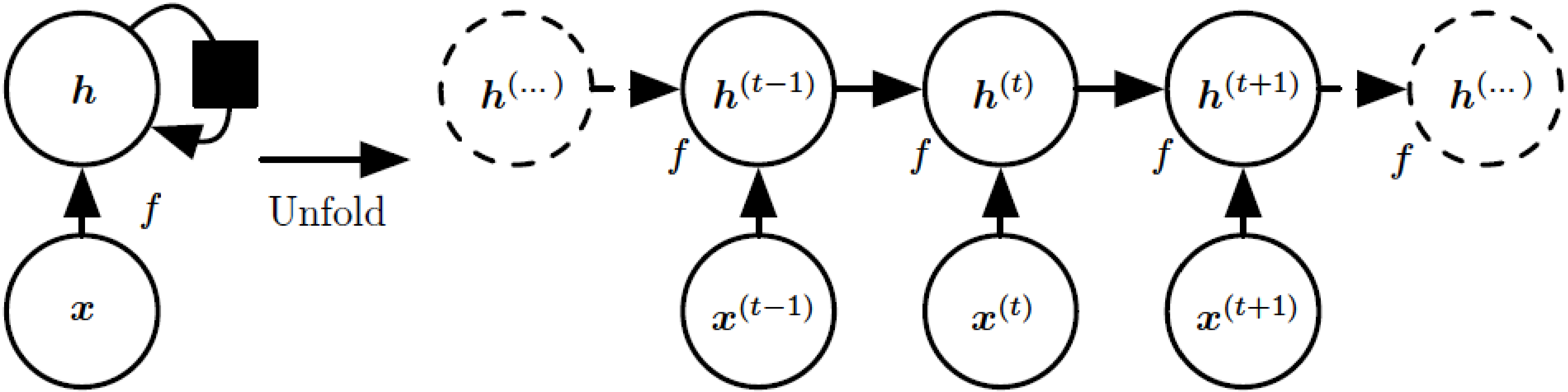
Elman RNN: external input

$$\boldsymbol{h}^{(t)} = f(\boldsymbol{h}^{(t-1)}, \boldsymbol{x}^{(t)}; \boldsymbol{\theta})$$



Elman RNN: external input

$$h^{(t)} = f(h^{(t-1)}, x^{(t)}; \theta)$$



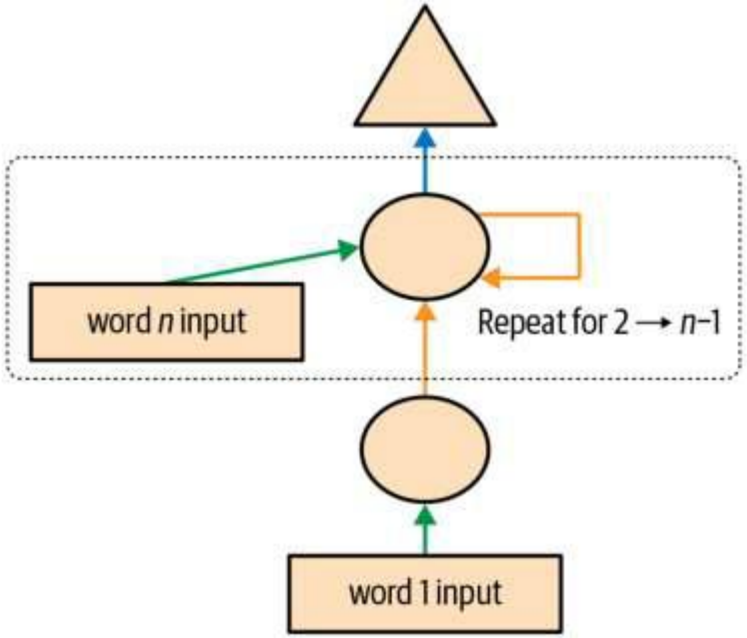
Elman RNN: single transition function

$$\boldsymbol{h}^{(t)} = g^{(t)}(\boldsymbol{x}^{(t)}, \boldsymbol{x}^{(t-1)}, \boldsymbol{x}^{(t-2)}, \ldots, \boldsymbol{x}^{(2)}, \boldsymbol{x}^{(1)})$$

vs

$$\boldsymbol{h}^{(t)} = f(\boldsymbol{h}^{(t-1)}, \boldsymbol{x}^{(t)}; \boldsymbol{\theta})$$

Elman RNN



Elman RNN

$$\begin{aligned}
\boldsymbol{x}_m &\triangleq \phi_{w_m} \\
\boldsymbol{h}_m &= \mathrm{RNN}(\boldsymbol{x}_m, \boldsymbol{h}_{m-1}) \\
\mathrm{p}(w_{m+1} \mid w_1, w_2, \ldots, w_m) &= \frac{\exp(\boldsymbol{\beta}_{w_{m+1}} \cdot \boldsymbol{h}_m)}{\sum_{w' \in \mathcal{V}} \exp(\boldsymbol{\beta}_{w'} \cdot \boldsymbol{h}_m)}
\end{aligned}$$

Elman RNN

$$\boldsymbol{x}_m \triangleq \phi_{w_m}$$

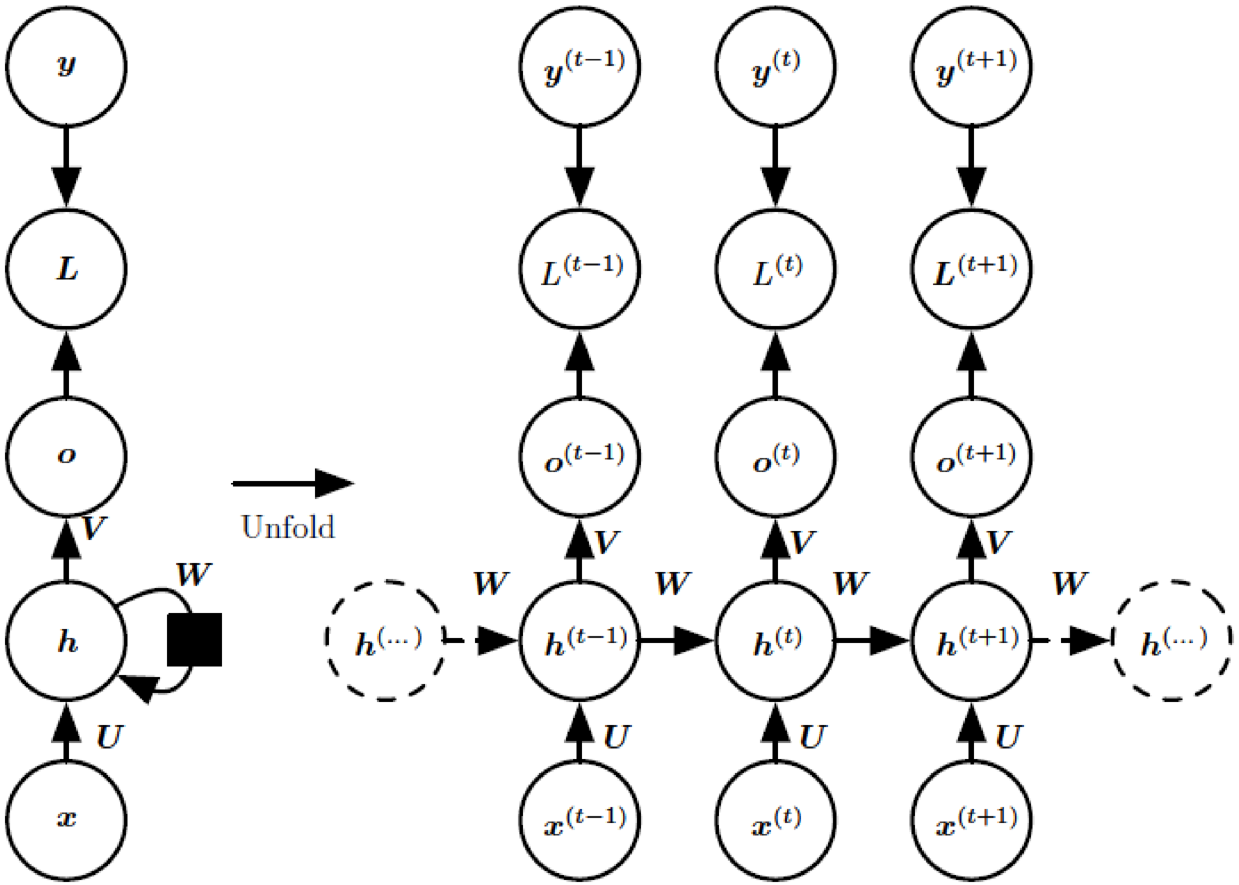
$$\boldsymbol{h}_m = \mathrm{RNN}(\boldsymbol{x}_m, \boldsymbol{h}_{m-1})$$

$$\mathrm{p}(w_{m+1} \mid w_1, w_2, \ldots, w_m) = \frac{\exp(\boldsymbol{\beta}_{w_{m+1}} \cdot \boldsymbol{h}_m)}{\sum_{w' \in \mathcal{V}} \exp(\boldsymbol{\beta}_{w'} \cdot \boldsymbol{h}_m)}$$

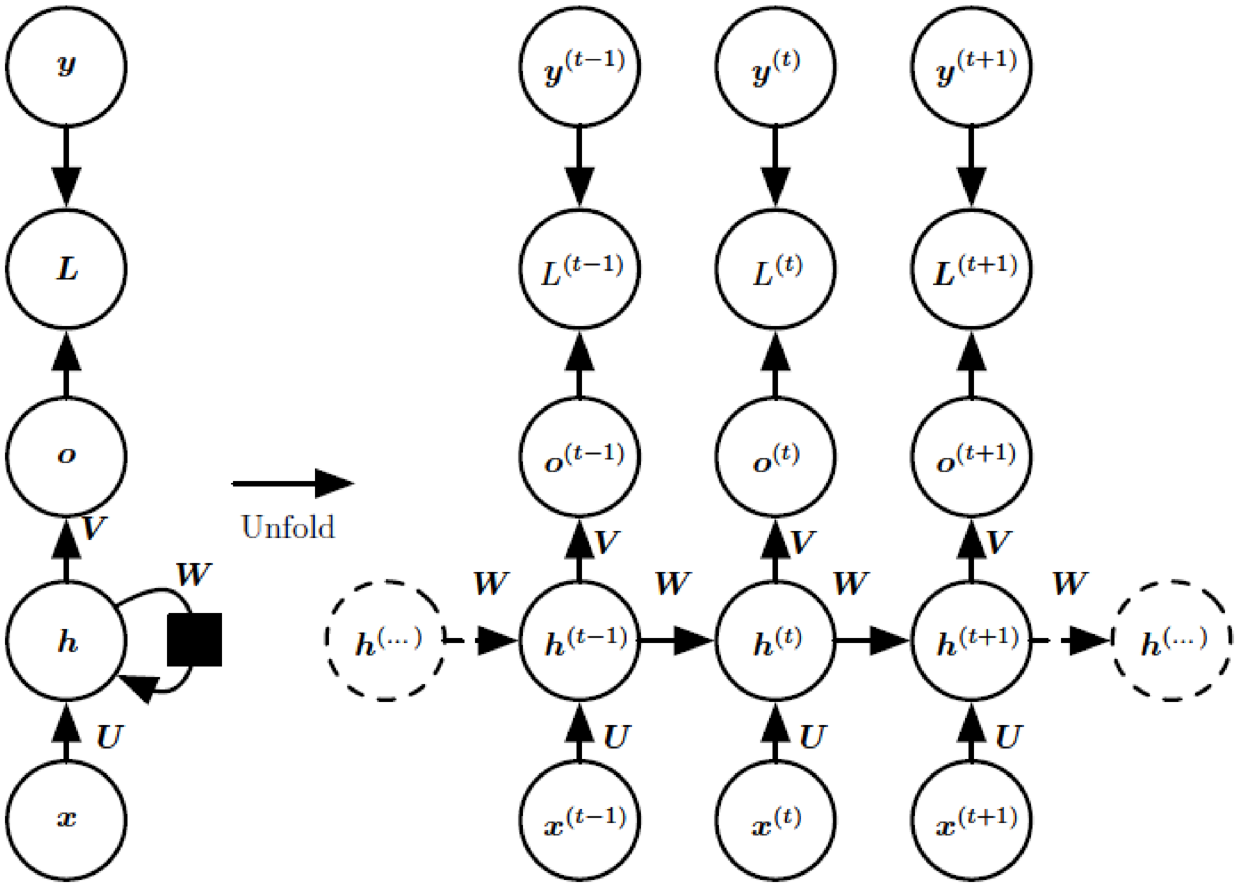
originally: $\mathrm{RNN}(\boldsymbol{x}_m, \boldsymbol{h}_{m-1}) \triangleq g(\boldsymbol{\Theta}\boldsymbol{h}_{m-1} + \boldsymbol{x}_m)$

more generally: (PyTorch notation) $g(\mathbf{W}_{ih}\mathbf{x}_m + \mathbf{b}_{ih} + \mathbf{W}_{hh}\mathbf{h}_{m-1} + \mathbf{b}_{hh})$

example of RNN

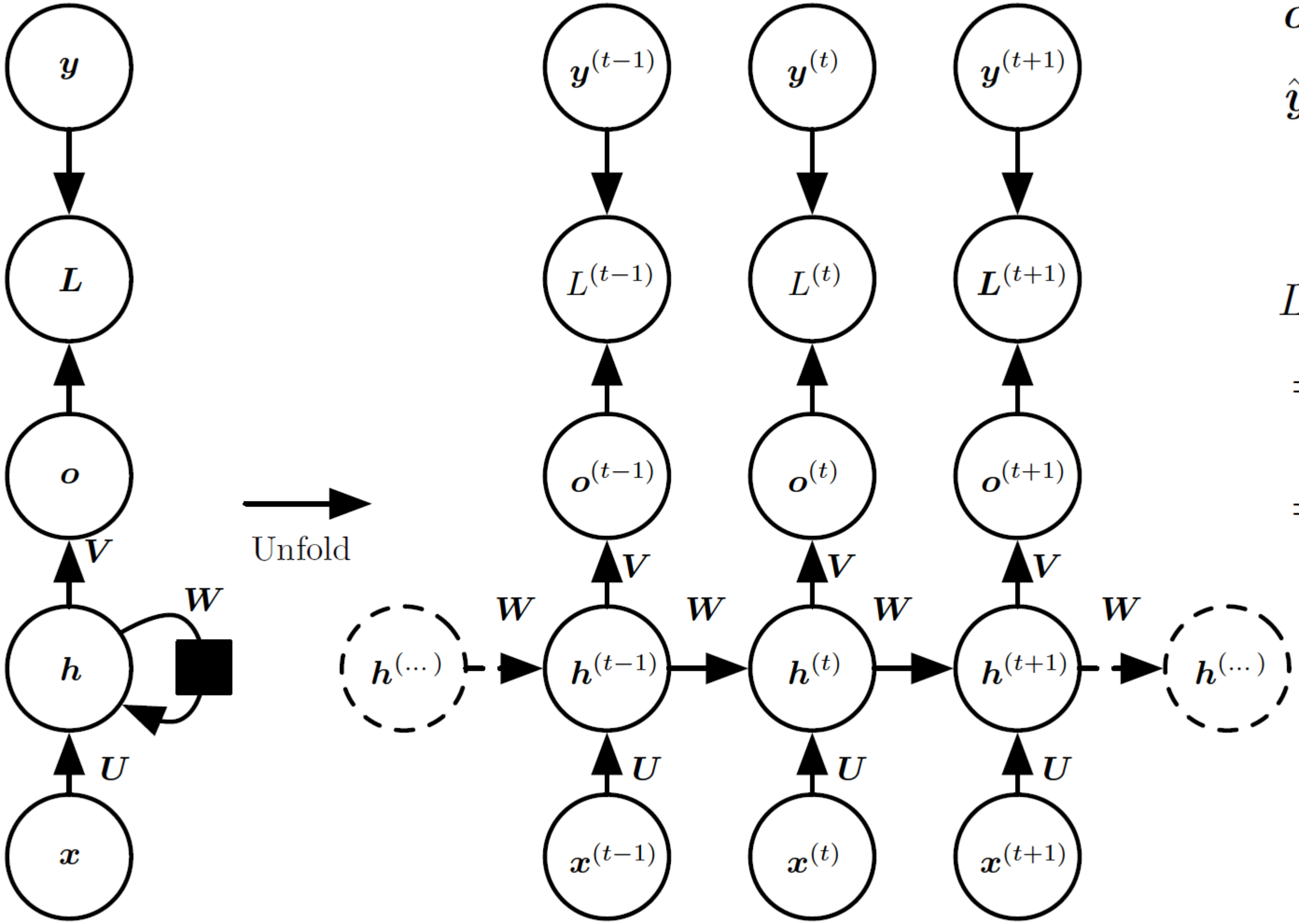


example of RNN



$$
\begin{aligned}
\boldsymbol{a}^{(t)} &= \boldsymbol{b} + \boldsymbol{W}\boldsymbol{h}^{(t-1)} + \boldsymbol{U}\boldsymbol{x}^{(t)}, \\
\boldsymbol{h}^{(t)} &= \tanh(\boldsymbol{a}^{(t)}), \\
\boldsymbol{o}^{(t)} &= \boldsymbol{c} + \boldsymbol{V}\boldsymbol{h}^{(t)}, \\
\hat{\boldsymbol{y}}^{(t)} &= \operatorname{softmax}(\boldsymbol{o}^{(t)}),
\end{aligned}
$$

example

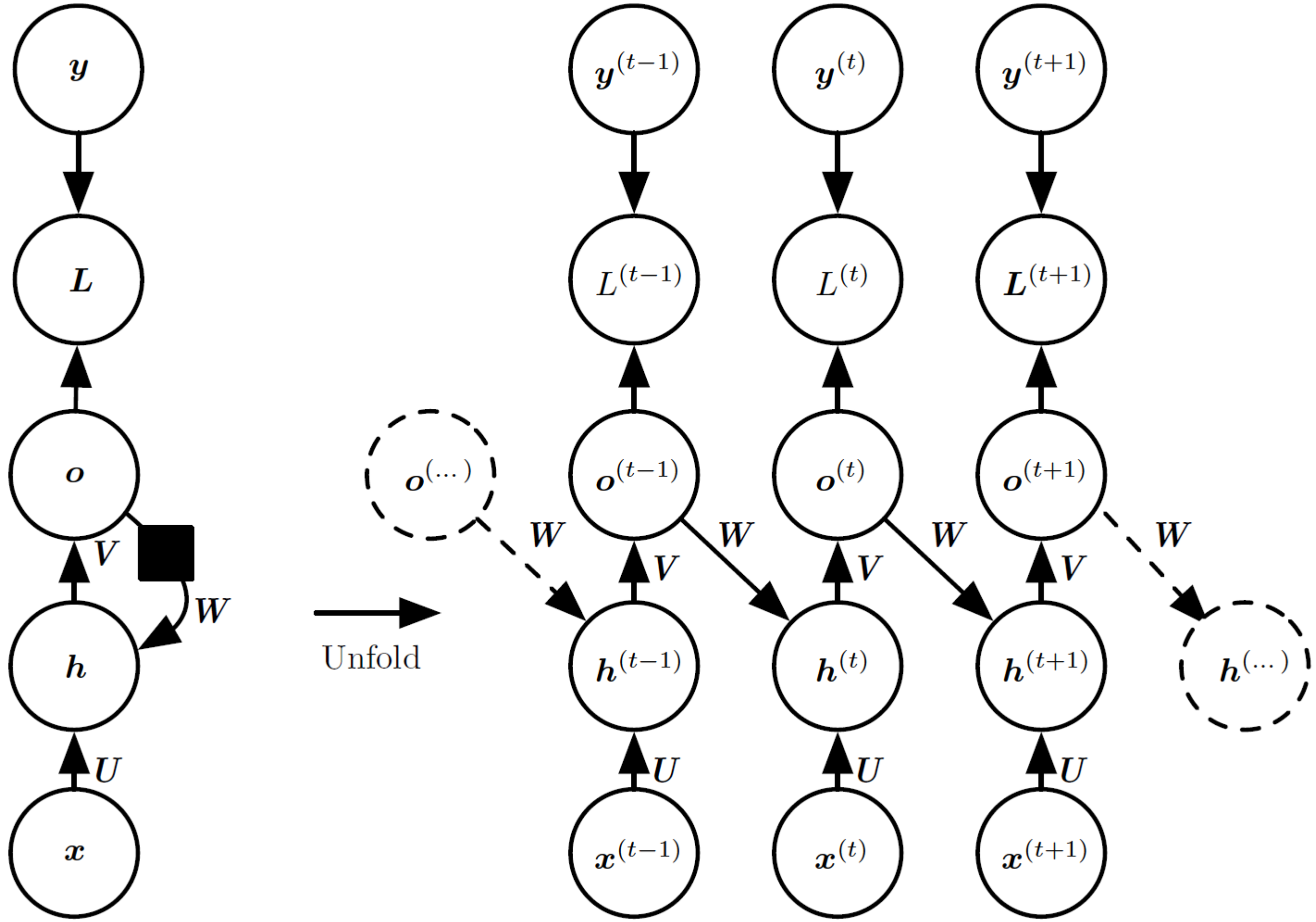


$$\boldsymbol{a}^{(t)} = \boldsymbol{b} + \boldsymbol{W}\boldsymbol{h}^{(t-1)} + \boldsymbol{U}\boldsymbol{x}^{(t)},$$
$$\boldsymbol{h}^{(t)} = \tanh(\boldsymbol{a}^{(t)}),$$
$$\boldsymbol{o}^{(t)} = \boldsymbol{c} + \boldsymbol{V}\boldsymbol{h}^{(t)},$$
$$\hat{\boldsymbol{y}}^{(t)} = \mathrm{softmax}(\boldsymbol{o}^{(t)}),$$

$$L\left(\{\boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(\tau)}\}, \{\boldsymbol{y}^{(1)}, \dots, \boldsymbol{y}^{(\tau)}\}\right)$$
$$= \sum_t L^{(t)}$$
$$= -\sum_t \log p_{\text{model}}\left(y^{(t)} \mid \{\boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(t)}\}\right)$$

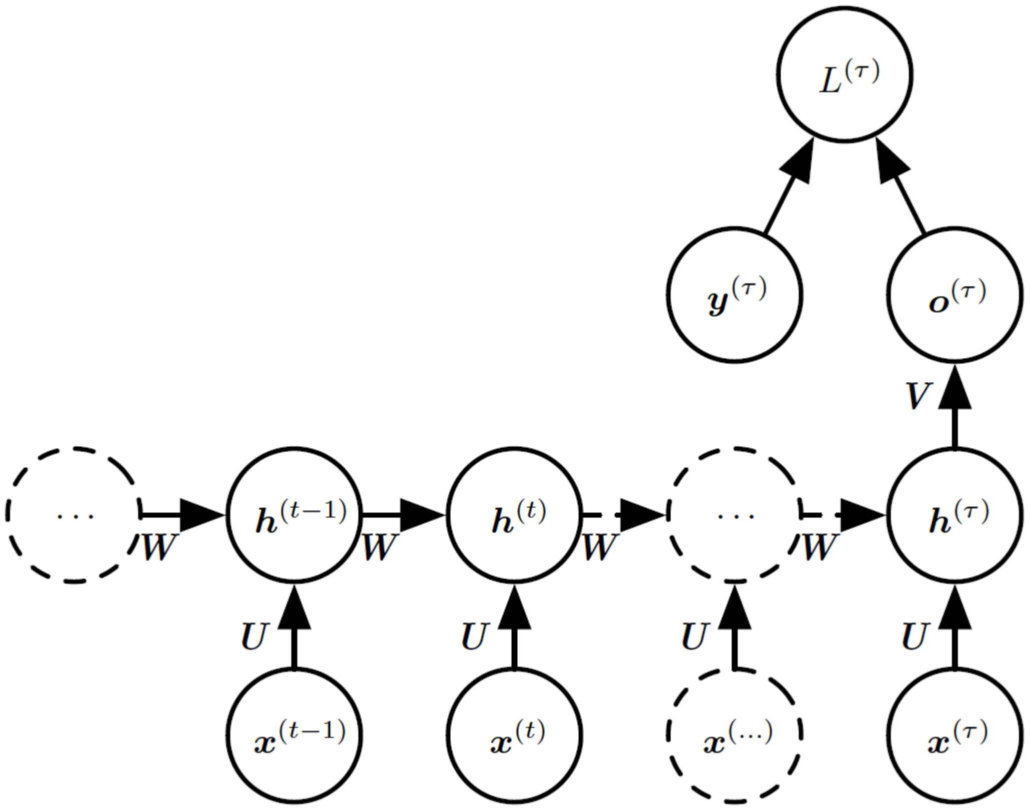
example

RNN that produces an output at each time step and has recurrent connections only from the output at one time step to the hidden units at the next time step

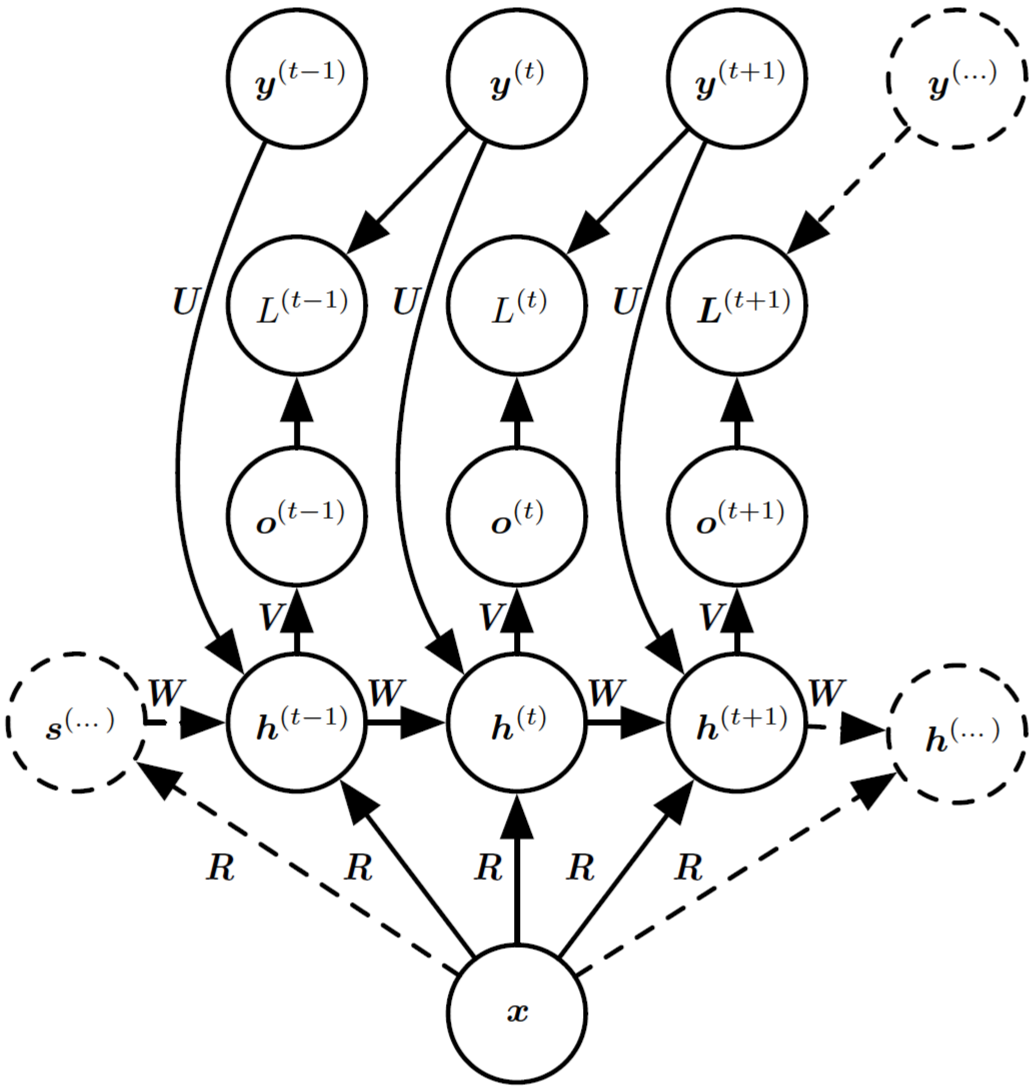


example

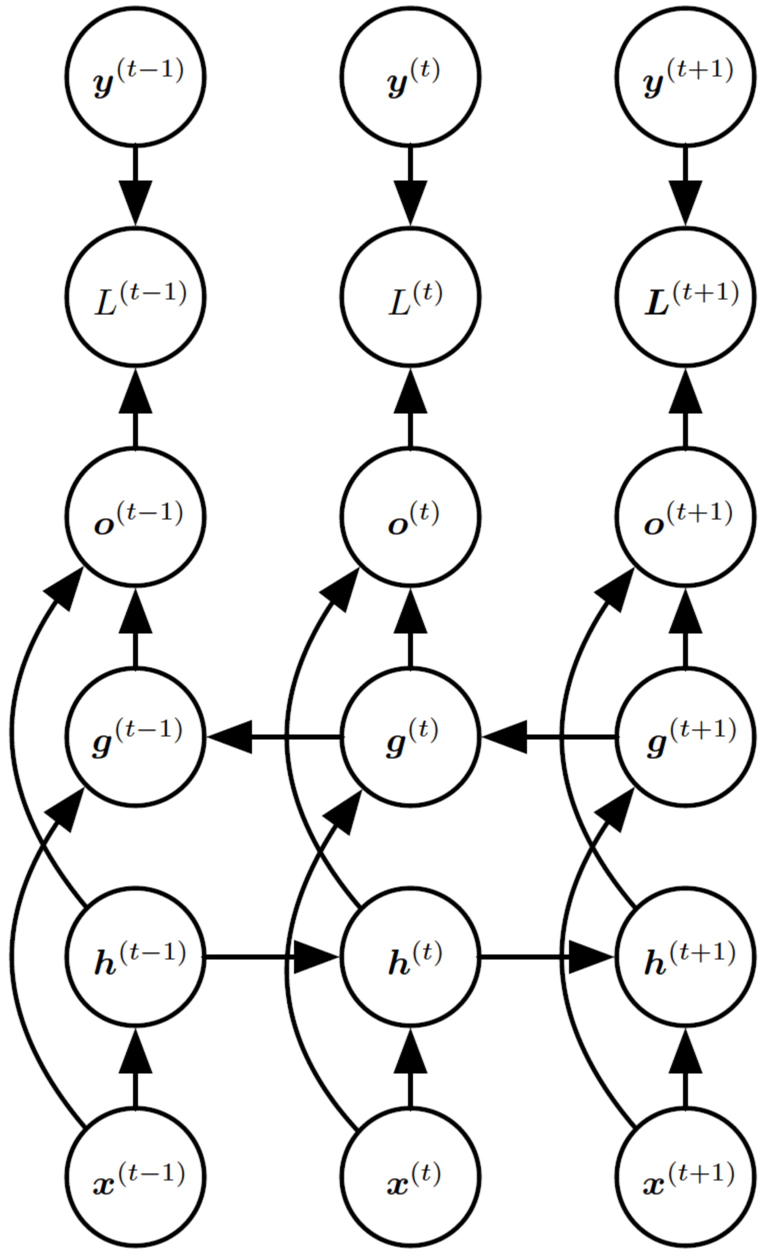
RNN with recurrent connections between hidden units, that reads an entire sequence and then produces a single output



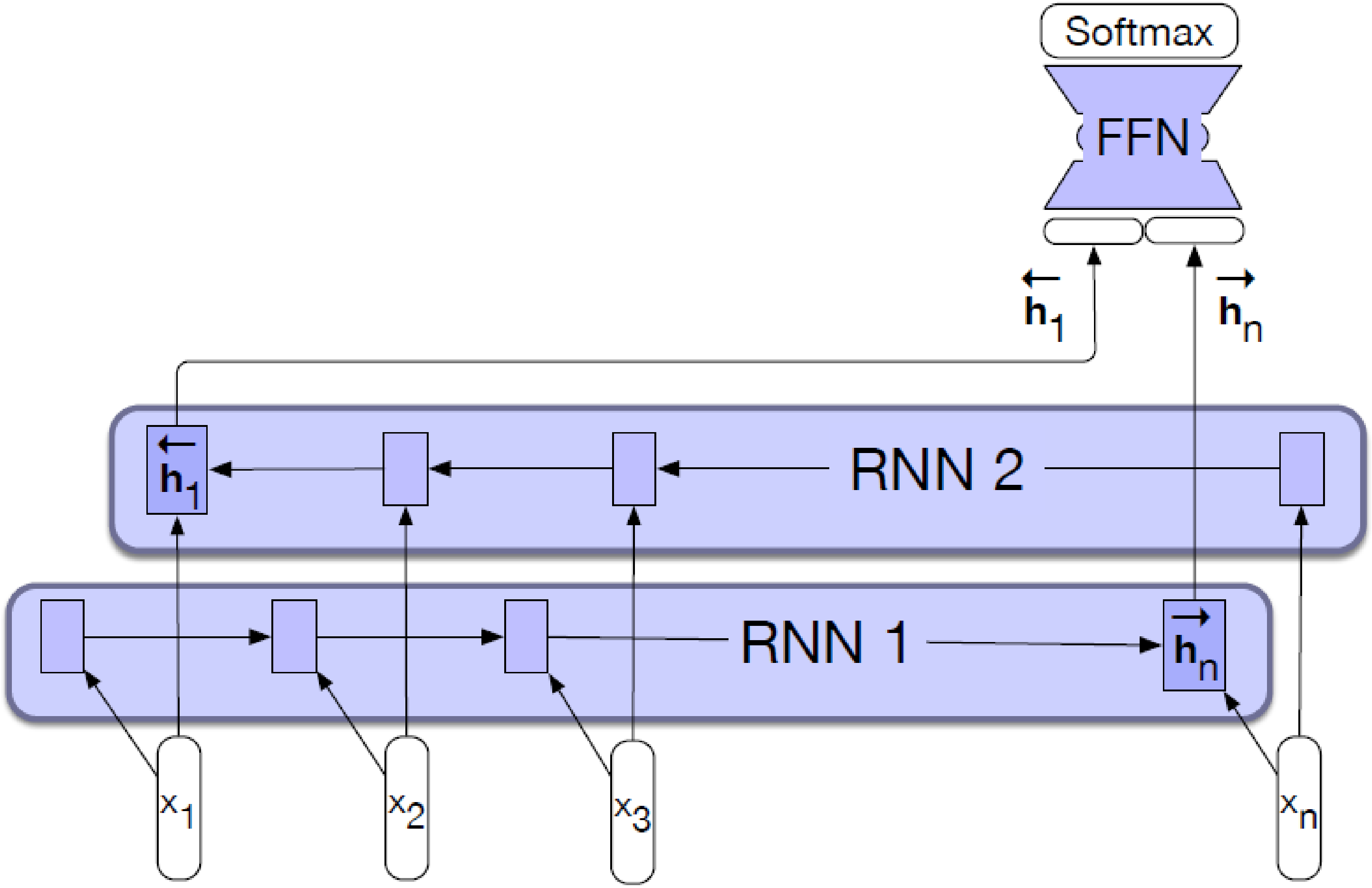
conditioning on inputs



bi-directional RNN



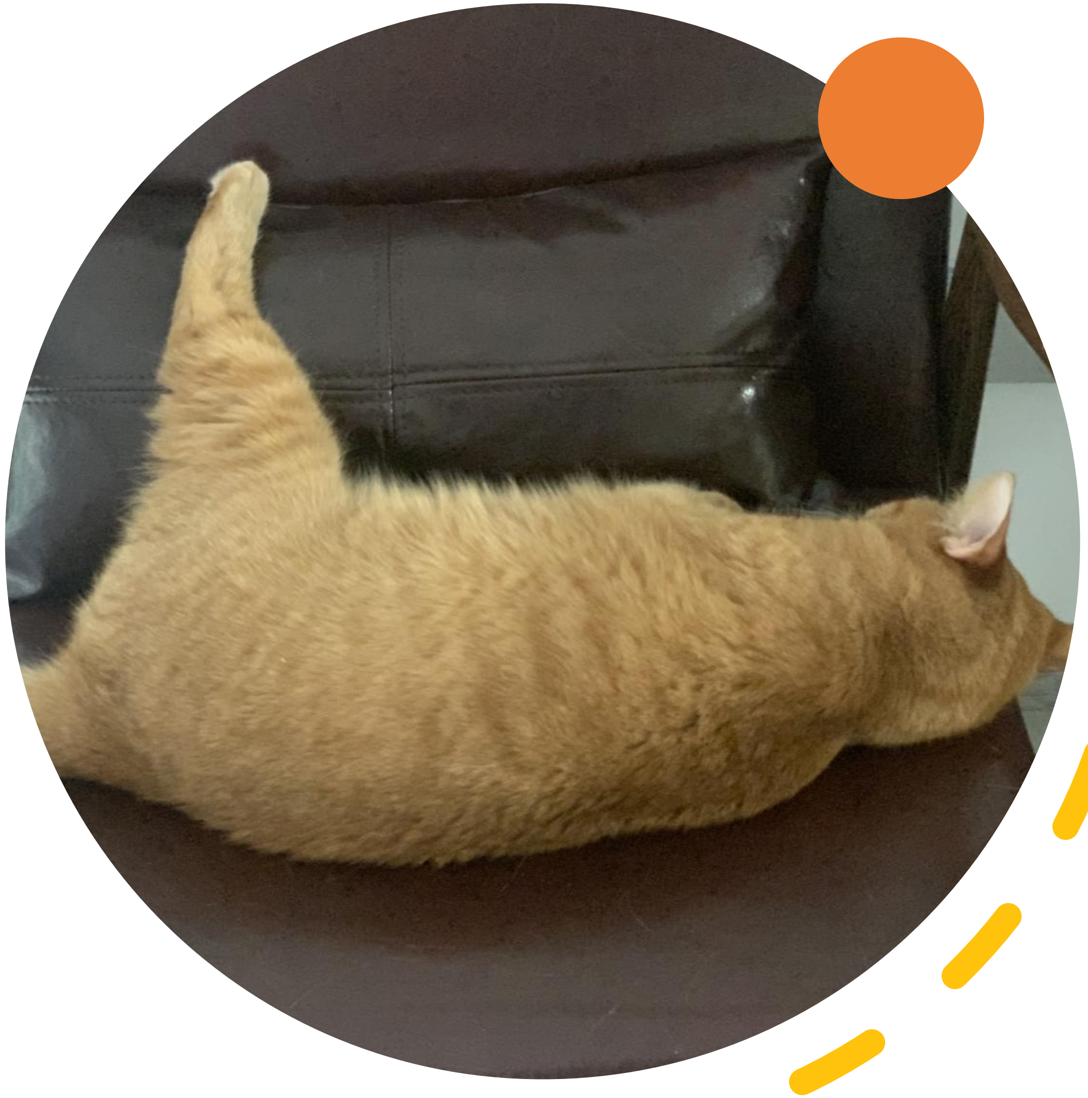
bi-directional RNN



Thank you!

Reference

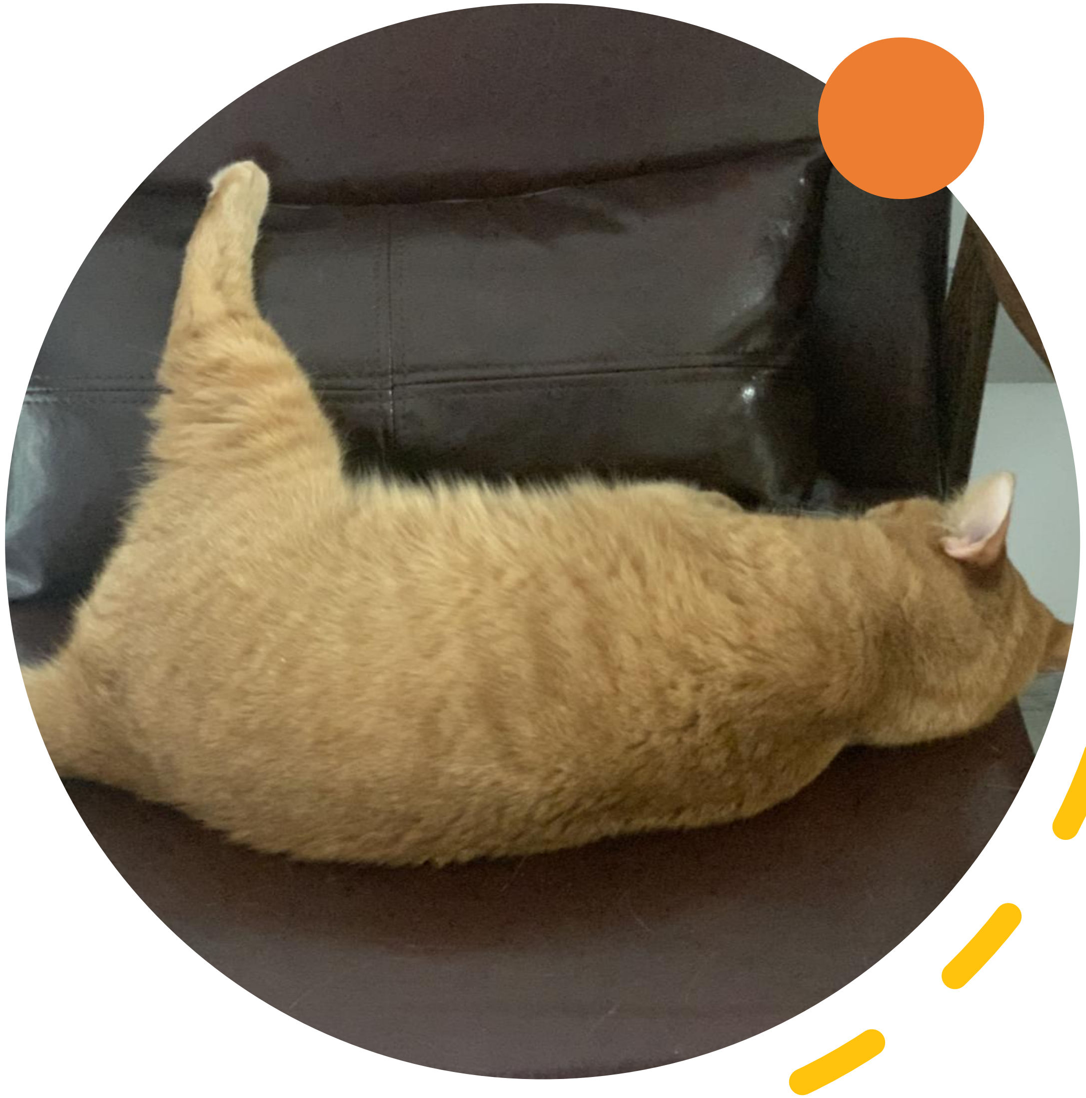
- Zeiler, M. D., & Fergus, R. (2014, September). *Visualizing and understanding convolutional networks*. In European conference on computer vision (pp. 818-833). Springer.
- Zhou, B., Lapedriza, A., Khosla, A., Oliva, A., & Torralba, A. (2017). *Places: A 10 million image database for scene recognition*. IEEE transactions on pattern analysis and machine intelligence, 40(6), 1452-1464.
- Murray, N., Marchesotti, L., & Perronnin, F. (2012, June). *AVA: A large-scale database for aesthetic visual analysis*. In 2012 IEEE Conference on Computer Vision and Pattern Recognition (pp. 2408-2415). IEEE.
- Cao, K., Rong, Y., Li, C., Tang, X., & Loy, C. C. (2018). *Pose-robust face recognition via deep residual equivariant mapping*. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 5187-5196).
- Jourabloo, A., Liu, Y., & Liu, X. (2018*). Face de-spoofing: Anti-spoofing via noise modeling*. In Proceedings of the European Conference on Computer Vision (ECCV) (pp. 290-306).
- Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). *Rich feature hierarchies for accurate object detection and semantic segmentation*. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 580-587).
- Girshick, R. (2015). *Fast R-CNN*. In Proceedings of the IEEE international conference on computer vision (pp. 1440-1448).



Thank you!

Reference

- Ren, S., He, K., Girshick, R., & Sun, J. (2015). *Faster R-CNN: Towards real-time object detection with region proposal networks*. Advances in neural information processing systems, 28, 91-99.
- Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). *You only look once: Unified, real-time object detection*. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 779-788).
- Redmon, J., & Farhadi, A. (2018). *Yolov3: An incremental improvement*. arXiv preprint arXiv:1804.02767.
- Kae, A., Sohn, K., Lee, H., & Learned-Miller, E. (2013). *Augmenting CRFs with Boltzmann machine shape priors for image labeling*. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2019-2026).
- Chen, L. C., Papandreou, G., Kokkinos, I., Murphy, K., & Yuille, A. L. (2017). *Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs*. IEEE transactions on pattern analysis and machine intelligence, 40(4), 834-848.
- Ma, L., Jia, X., Sun, Q., Schiele, B., Tuytelaars, T., & Van Gool, L. (2017). *Pose guided person image generation*. In Proceedings of the 31st International Conference on Neural Information Processing Systems (pp. 405-415).
- Zhu, J. Y., Park, T., Isola, P., & Efros, A. A. (2017). *Unpaired image-to-image translation using cycle-consistent adversarial networks*. In Proceedings of the IEEE international conference on computer vision (pp. 2223-2232).
- Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., ... & Shi, W. (2017). *Photo-realistic single image super-resolution using a generative adversarial network*. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4681-4690).



Thank you!

Reference

- Ch 6, *Natural Language Processing* by Eisenstein.
- Ch 10, *Deep Learning*.
- Jurafsky, D., & Martin, J. H. (2018). *Speech and language processing (draft)*. https://web. stanford. edu/~ jurafsky/slp3.
- Goldberg, Y. (2016). A primer on neural network models for natural language processing. *Journal of Artificial Intelligence Research*, 57, 345-420.