Deep Learning Theory and Applications

RNNs and LSTMs

Yale

CPSC/AMTH 452/552

CBB 663

Natural Language Processing

- Field of AI concerned with interactions between humans and computers
- How to program computers to process and analyze natural language data (as opposed to computer languages)

NLP Tasks

- Speech recognition
- Natural language generation
- Natural language understanding

- More specific tasks
  - Machine translation
  - Sentiment Analysis
  - Question Answering
  - Text generation

Question

- Do CNNs work on data with time structure rather than space structure?

CNN work well on space structure eg. image, DNA seq

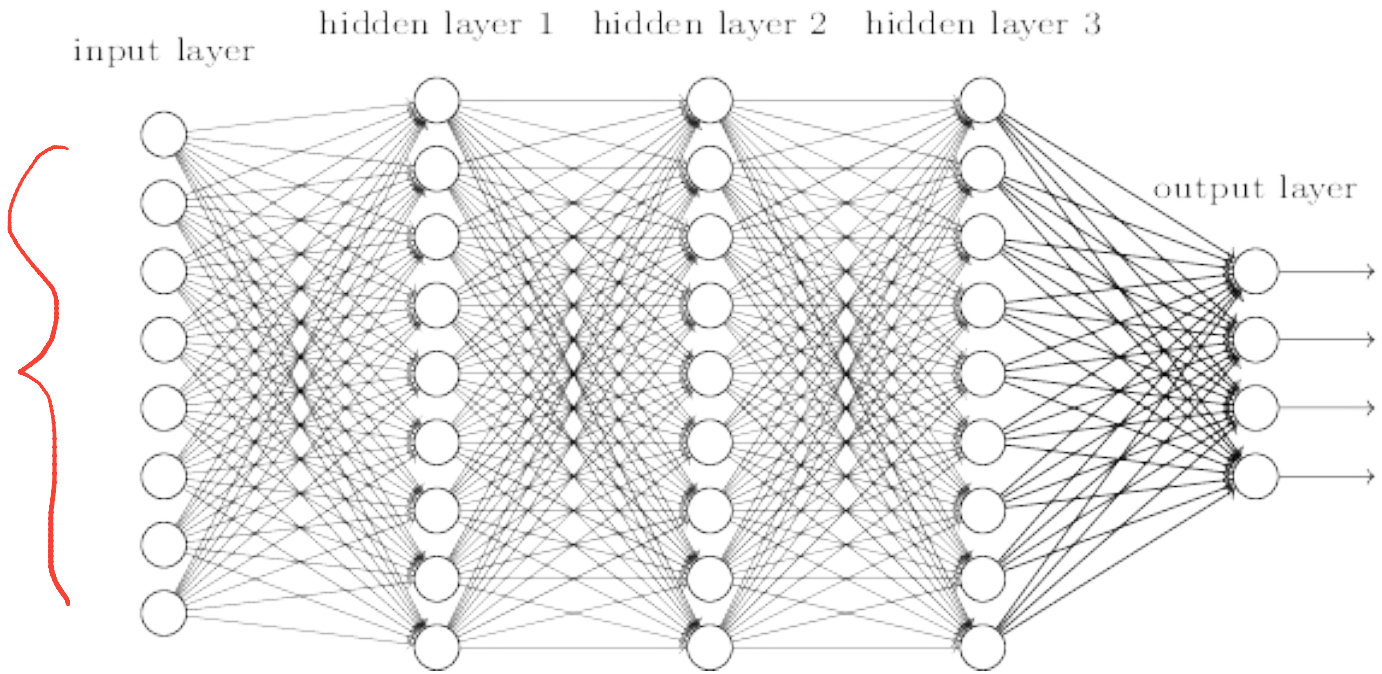
localized structure

CNN has fixed length structure

can't be used for variable lengths

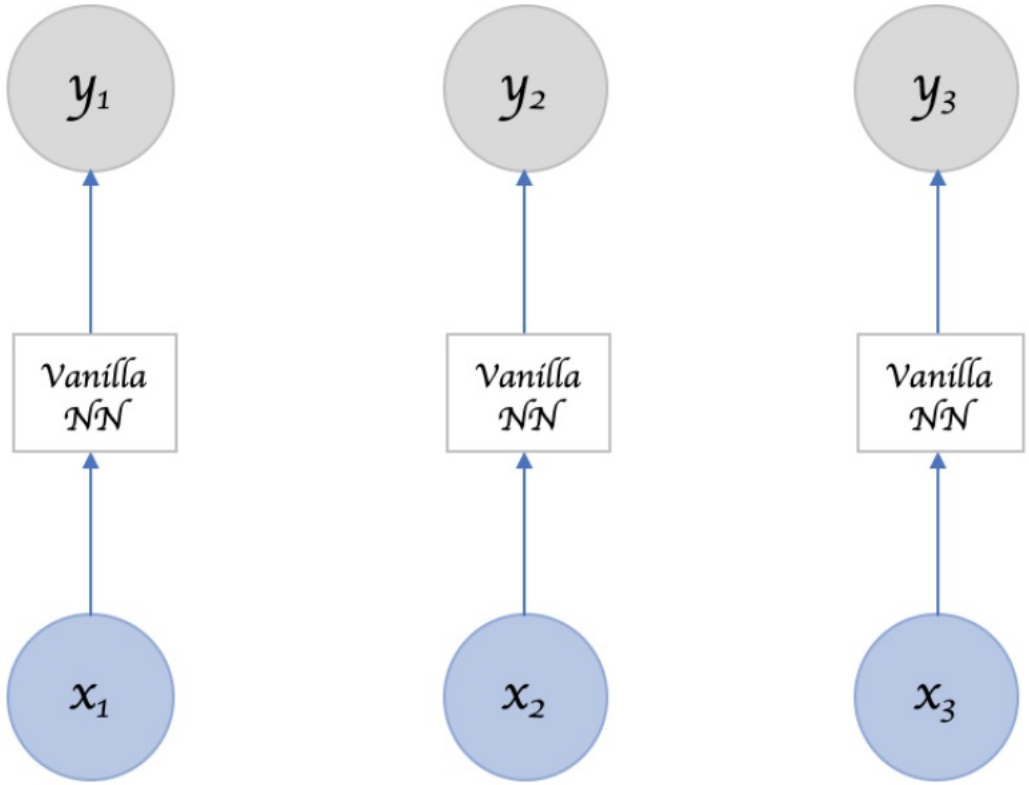
- What if the time-structure has variable lengths?

Fixed sized inputs



Normal neural networks have to have predetermined input size

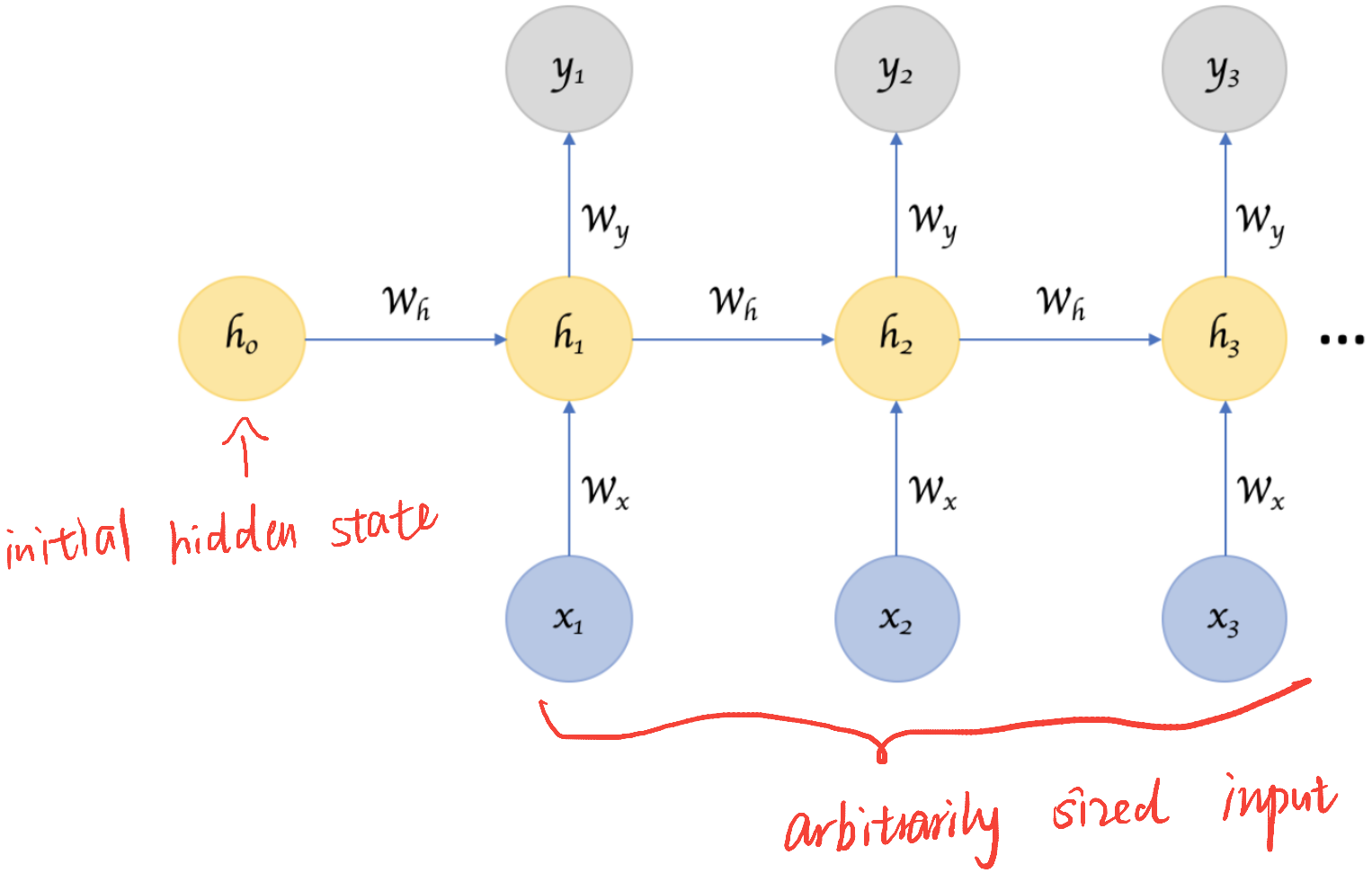
Contrast with calling NN repeatedly



Order and sequence matters

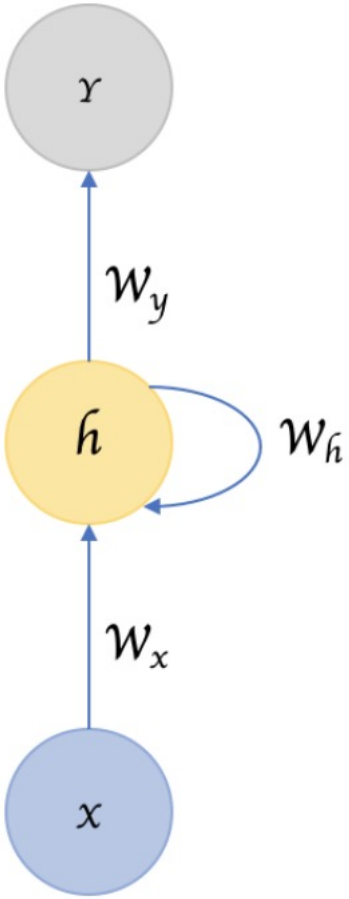
can't remember past

Recurrent Neural Networks

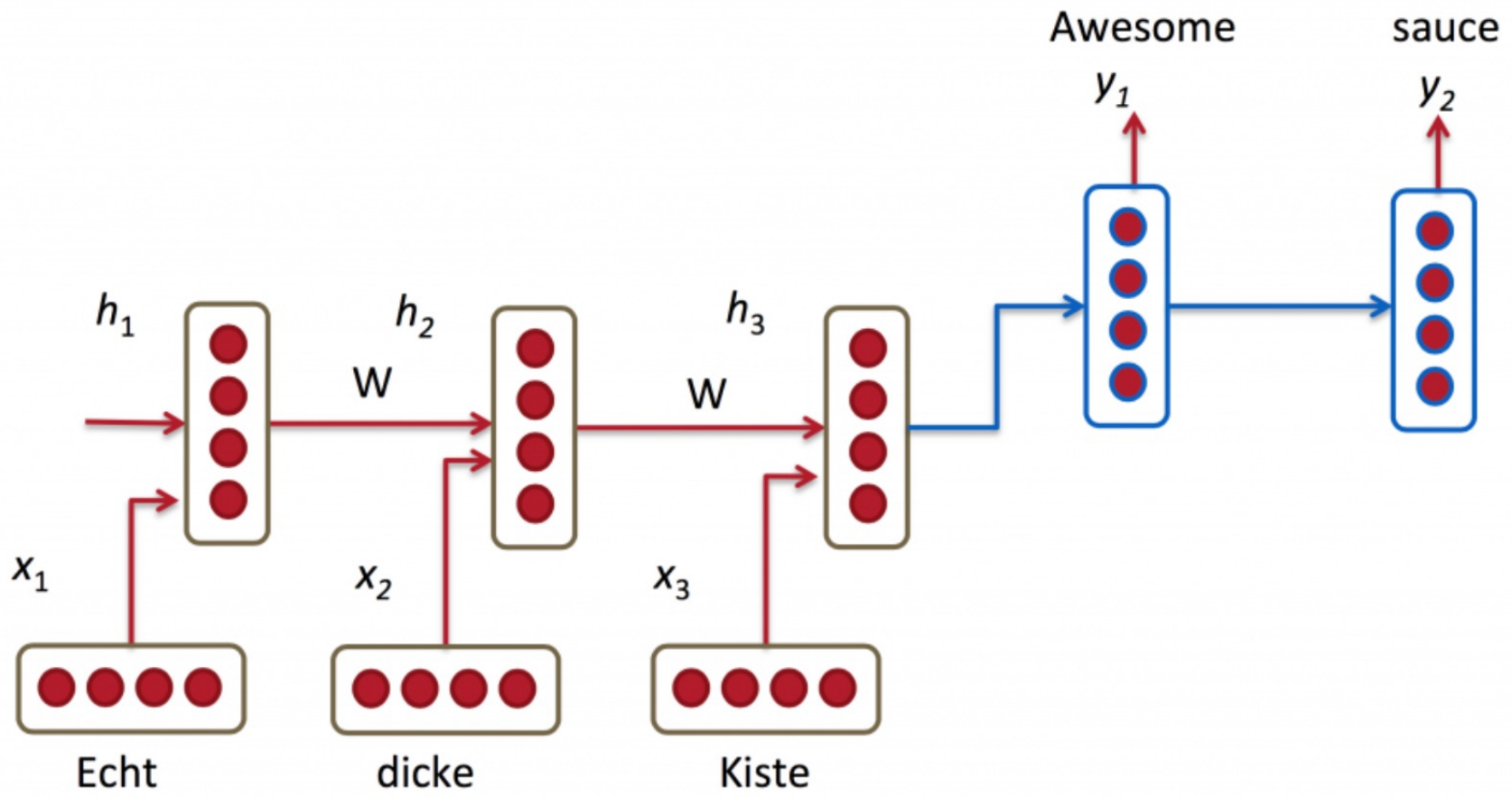


Takes arbitrarily sized inputs and "remember" a hidden state of information

Feed forward vs Self-Loops



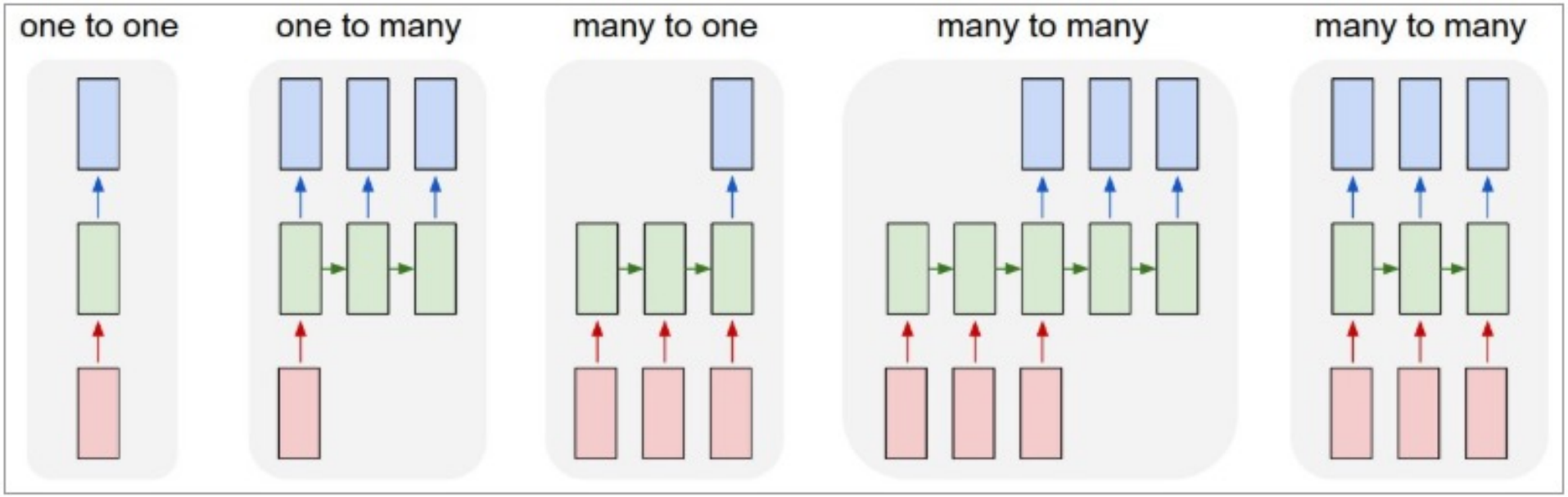
Processing Sequential Input



Sequential Newal Network

Sequence oriented tasks

classification image captioning sentiment analysis machine translation denoising



Variable length inputs, variable length outputs, variable length computation

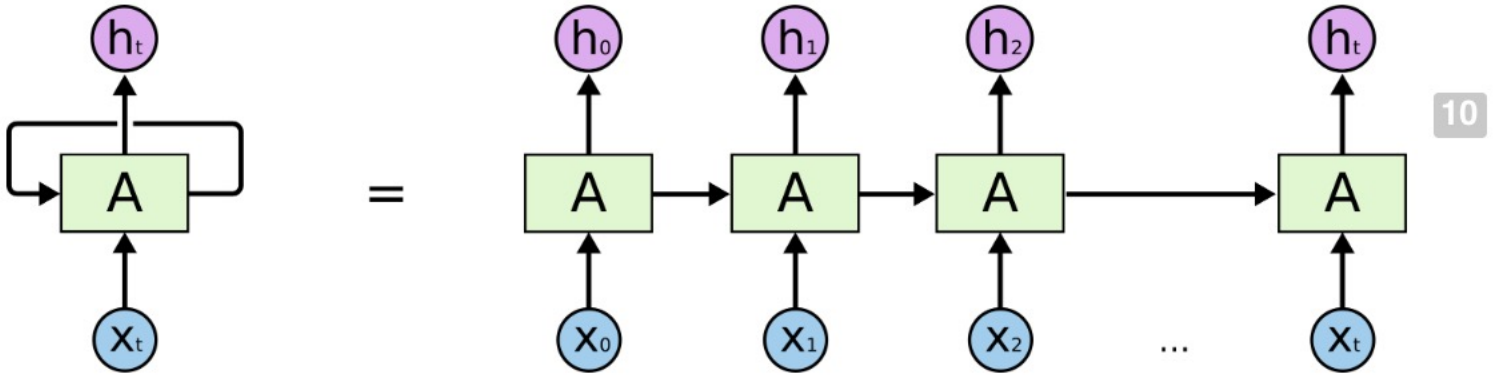
Backpropagation Through Time (BPTT)

similar to backpropagation

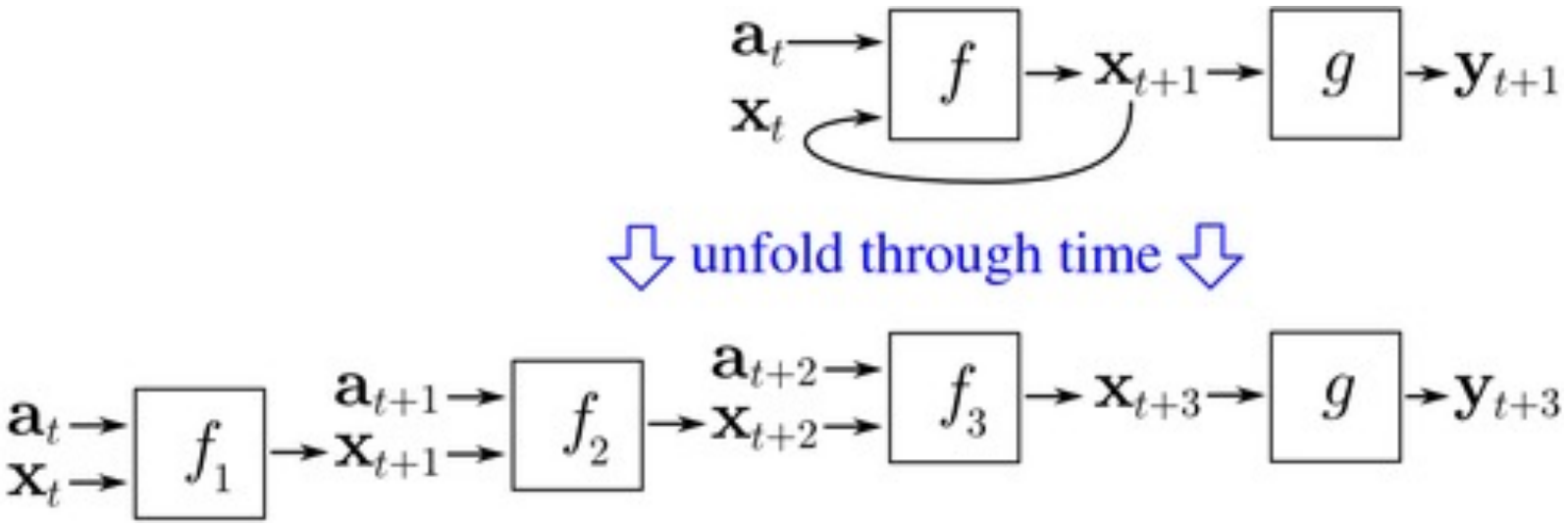
Difference: lay out several copies of NN, length of connection is a hyperparam

depends on expect your seq to be

longer or shorter



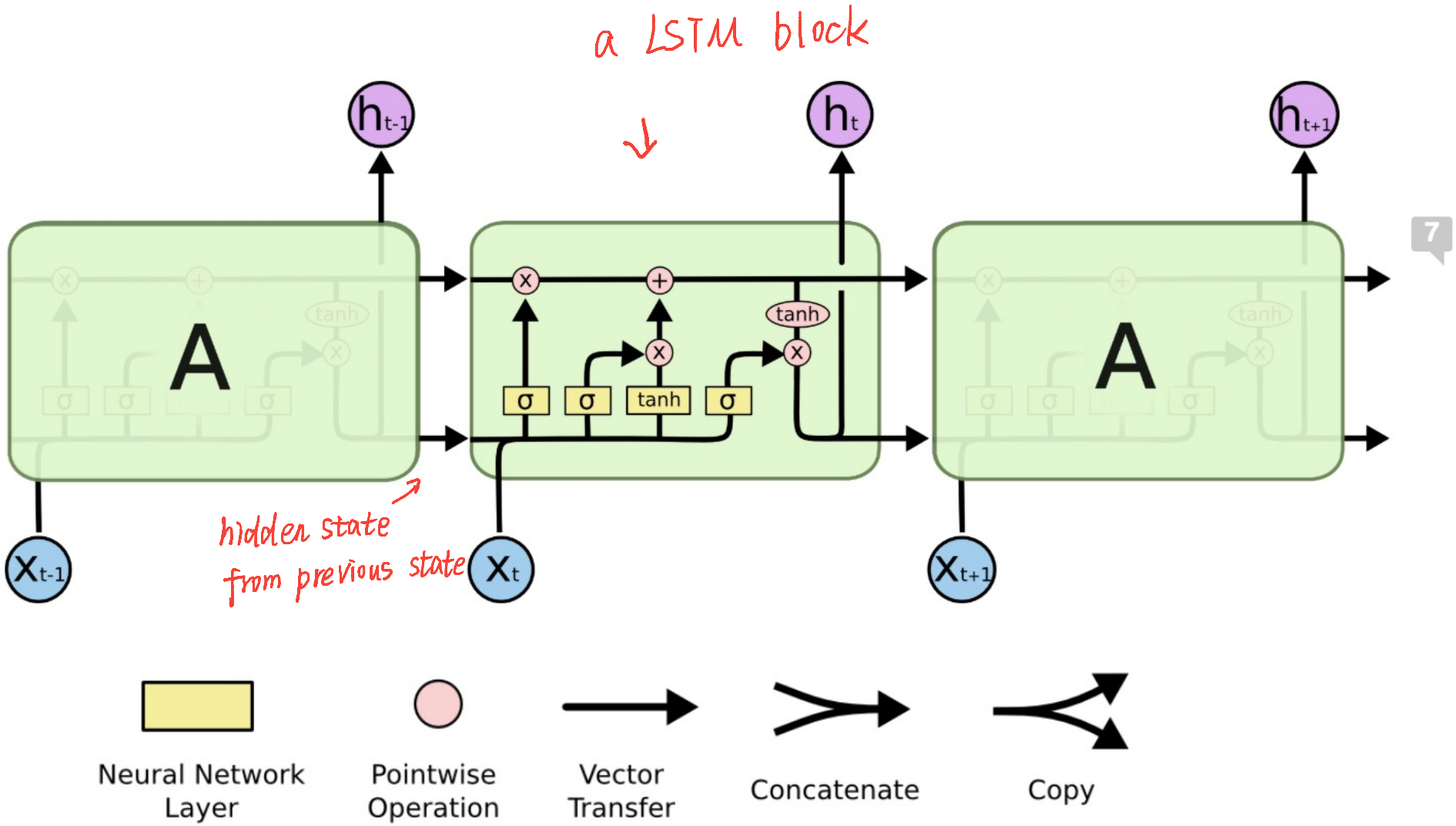
All unfoldings share parameters



BPTT Computes Gradients for Many Time Steps

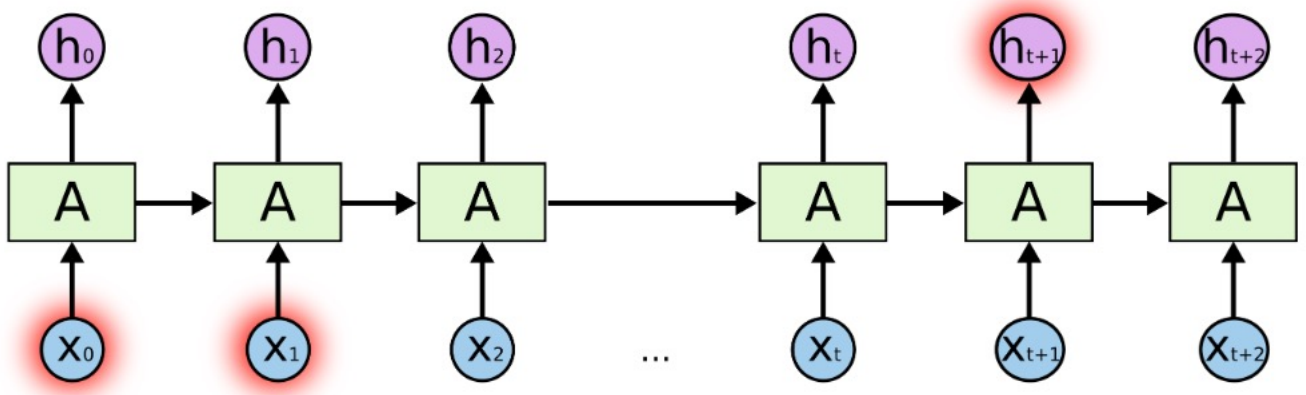
```
Back_Propagation_Through_Time(a, y)   // a[t] is the input at time t. y[t] is the output
    Unfold the network to contain k instances of f
    do until stopping criteria is met:
        x := the zero-magnitude vector // x is the current context
        for t from 0 to n − k do      // t is time. n is the length of the training sequence
            Set the network inputs to x, a[t], a[t+1], ..., a[t+k−1]
            p := forward-propagate the inputs over the whole unfolded network
            e := y[t+k] − p;           // error = target − prediction
            Back-propagate the error, e, back across the whole unfolded network
            Sum the weight changes in the k instances of f together.
            Update all the weights in f and g.
            x := f(x, a[t]);           // compute the context for the next time-step
```

Notation


a LSTM block
h_{t-1}
h_t
h_{t+1}
A
A
tanh
σ
σ
tanh
σ
hidden state
from previous state
X_{t-1}
X_t
X_{t+1}
Neural Network Layer
Pointwise Operation
Vector Transfer
Concatenate
Copy

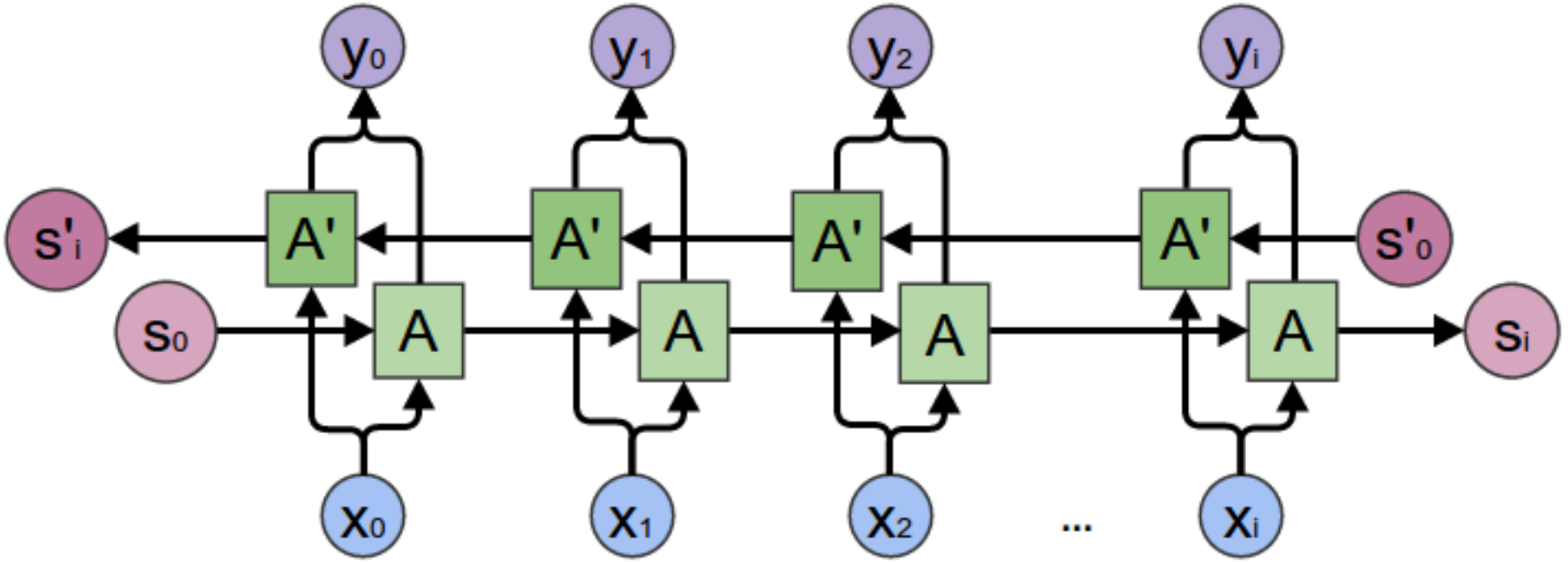
RNNs not good at handling long term dependencies

vanishing gradient problem



Bidirectional RNN

2 different RNNs { 1st RNN seq →
2nd RNN ←

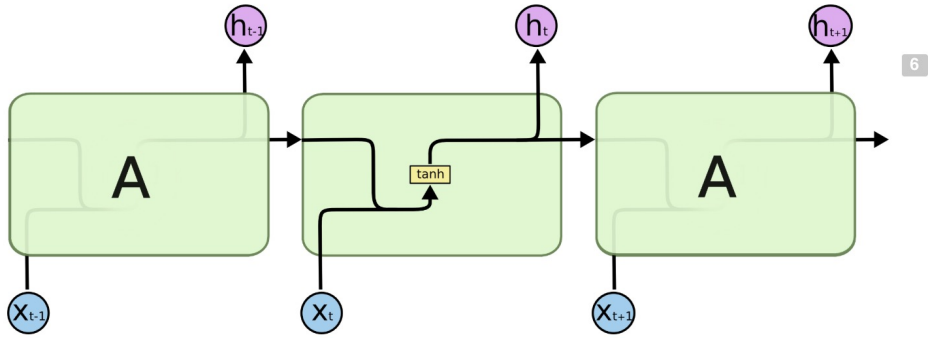


right → left

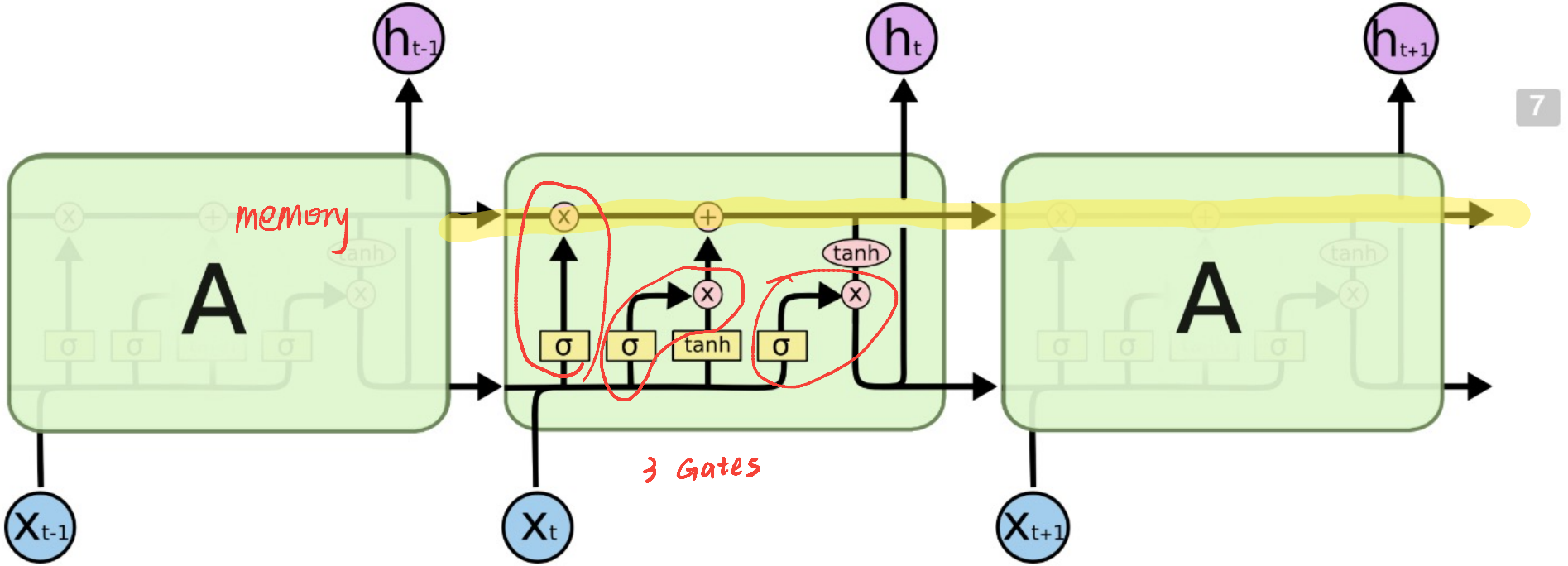
left → right

LSTMs

memory is like skip connection through time

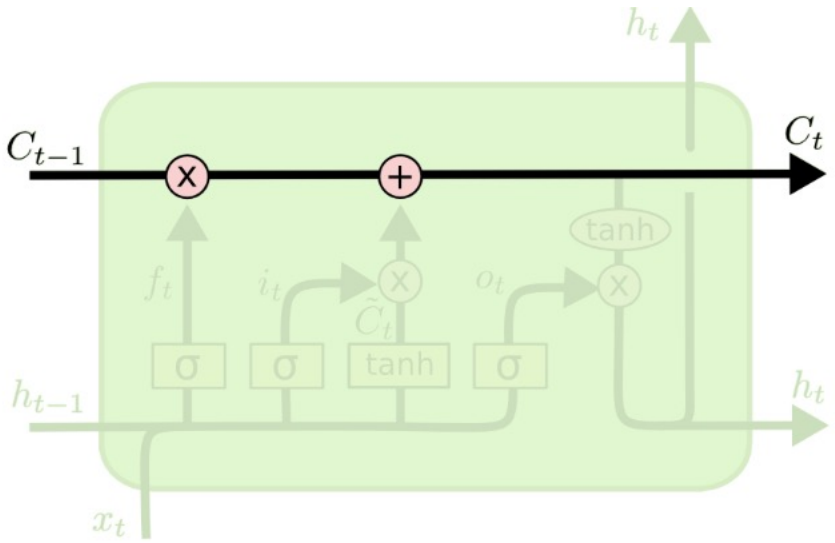


6



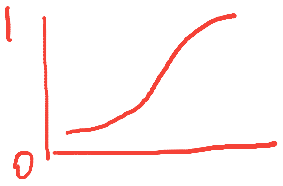
7

Running Cell State

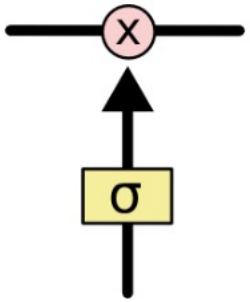


Gates used in GRU and LSTM

Gates modify hidden state



controls how much info go through from bottom to up

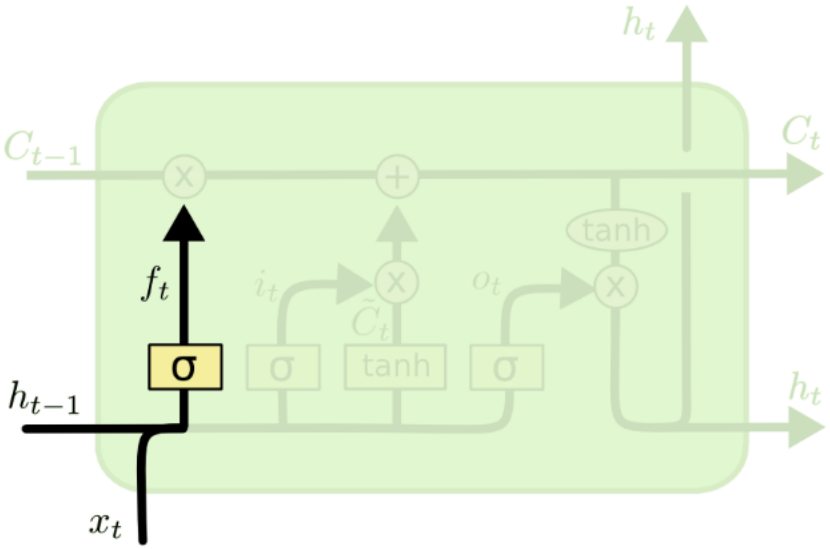


A gate is implemented as a sigmoidal layer that outputs a value between 0 and 1 and controls how much of an input should be let through

0=don't let anything through
1= let everything through

Indicates pointwise multiplication

Forget Gate

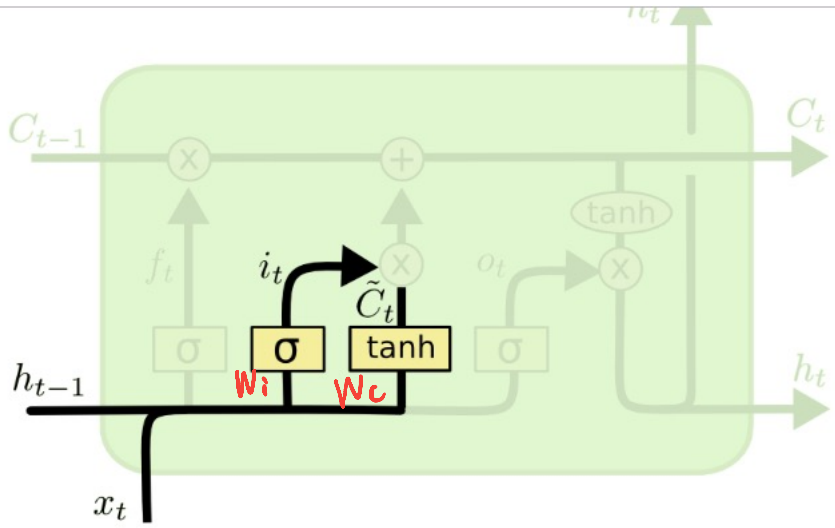


$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$

How much of the previous hidden state should be forgotten?

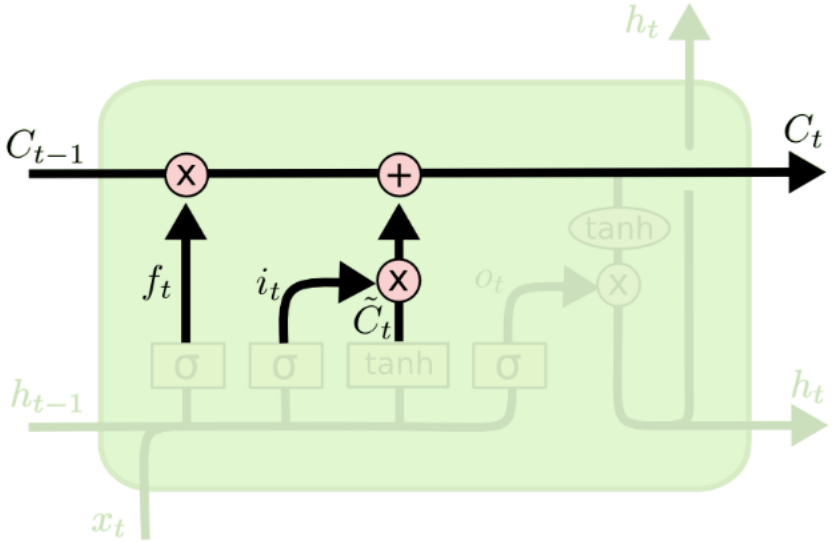
Wf determined

Creating New Hidden State



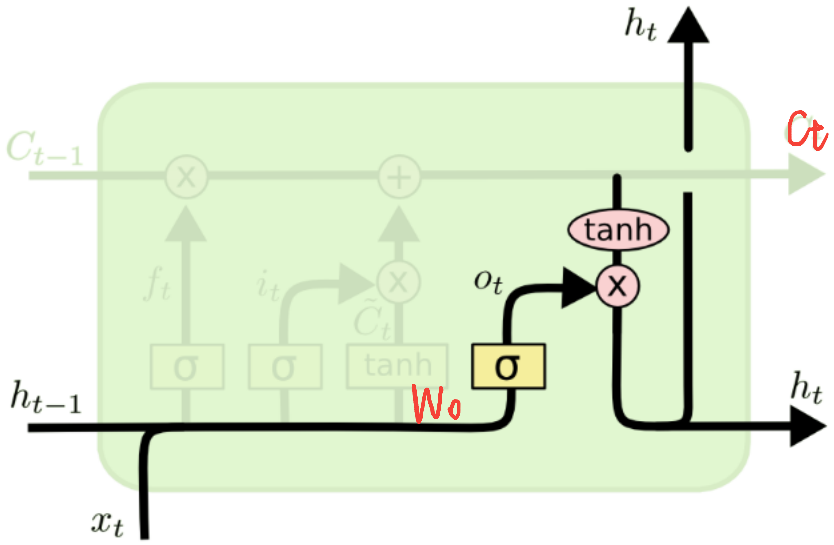
$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] \; + \; b_i)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

Output

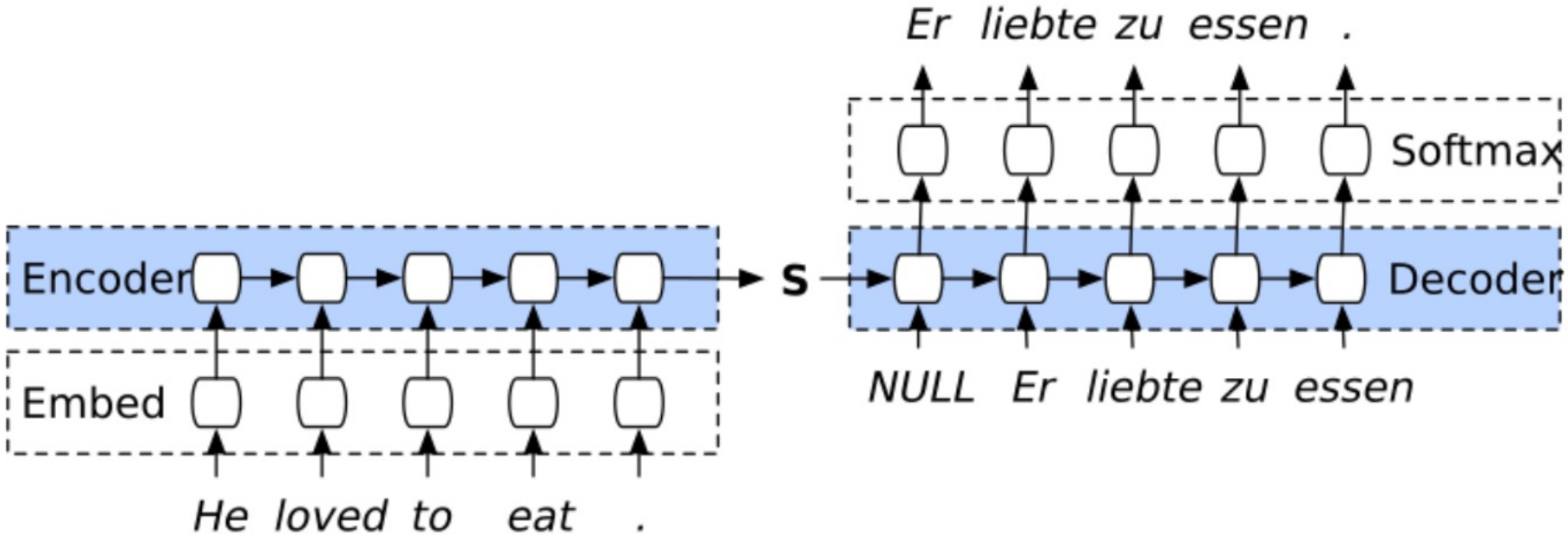


$$o_t = \sigma\left(W_o\left[h_{t-1}, x_t\right] + b_o\right)$$

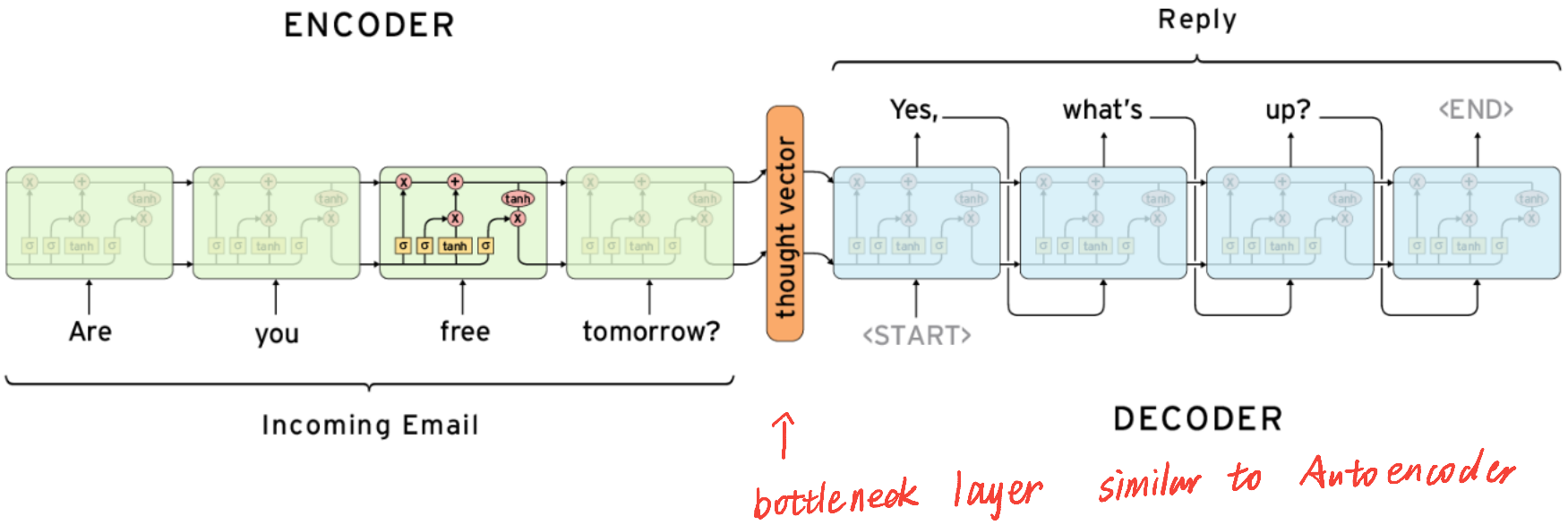
$$h_t = o_t * \tanh\left(C_t\right)$$

LSTM often used as

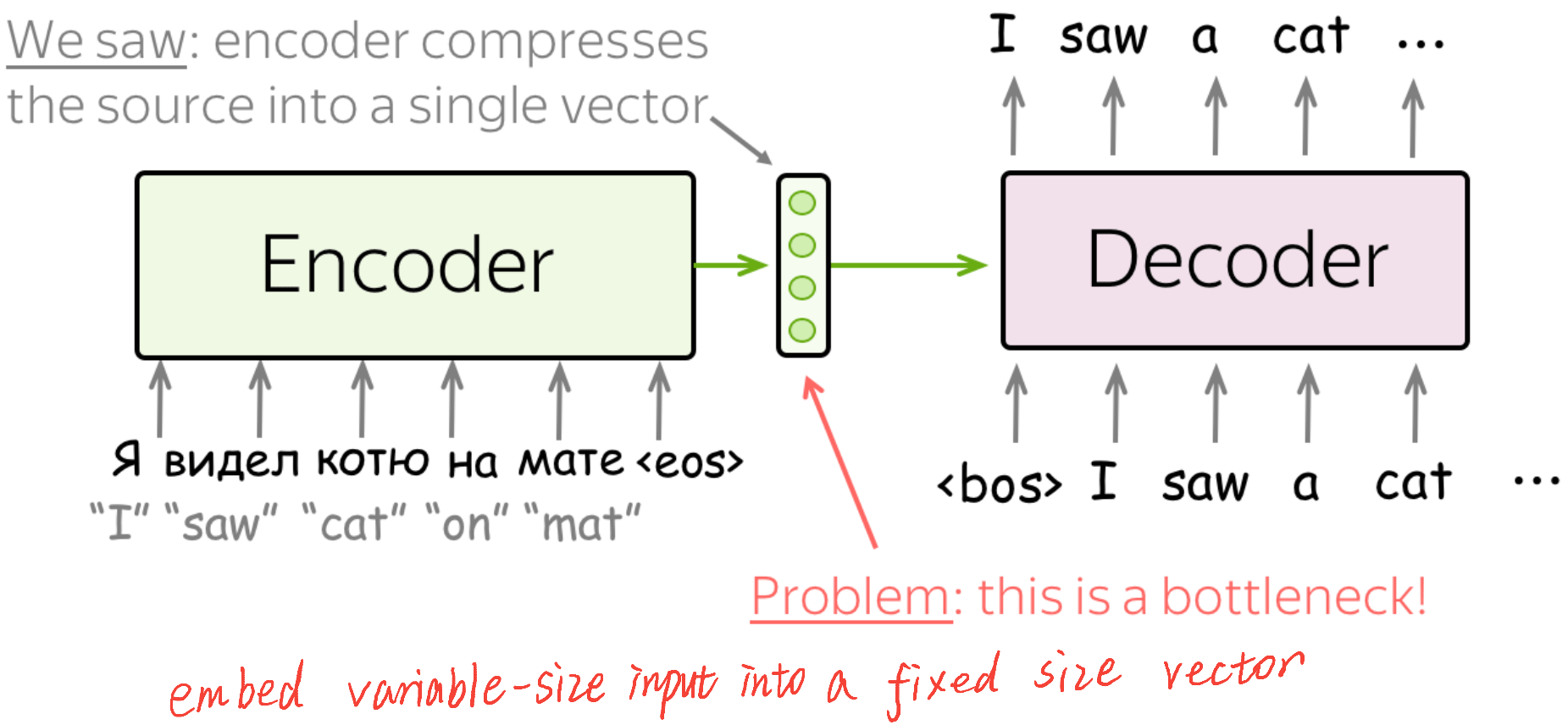
Sequence-to-sequence model



Implementation with LSTM

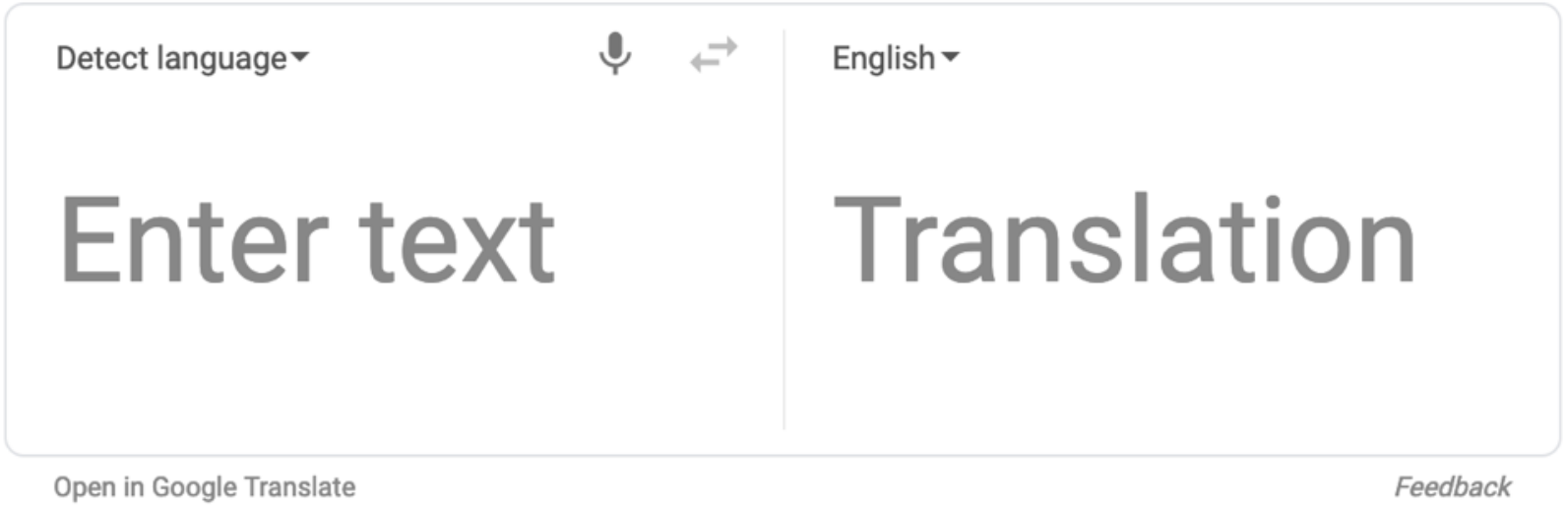


Problem


We saw: encoder compresses the source into a single vector
Encoder
Decoder
I saw a cat ...
Я видел котю на мате <eos>
"I" "saw" "cat" "on" "mat"
<bos> I saw a cat ...
Problem: this is a bottleneck!
embed variable-size input into a fixed size vector

Application

Machine Translation



Statistical machine translation:

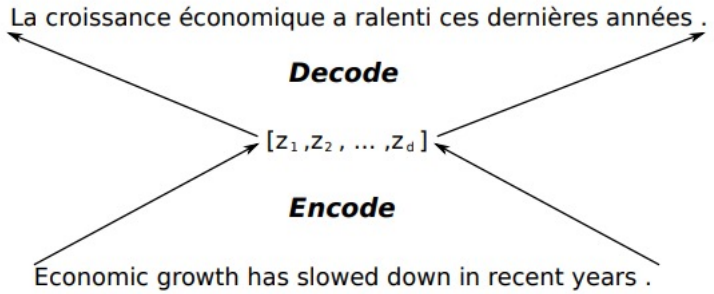
P(e|f) e = english phrase f= foreign phrase

$p(e|f) \propto p(f|e)p(e)$ language model $p(e)$

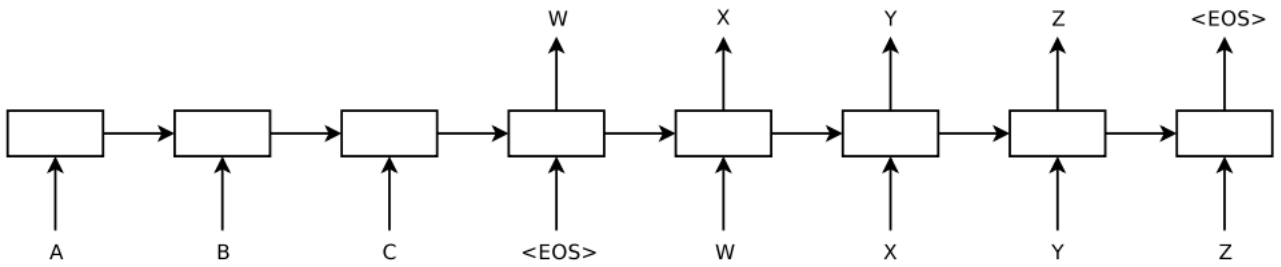
Translates n-grams phrases consisting of n-words

Neural Machine translation

Encoder-decoder architecture



Cho et al 2014, Sutskever 2014



Sentiment Analysis

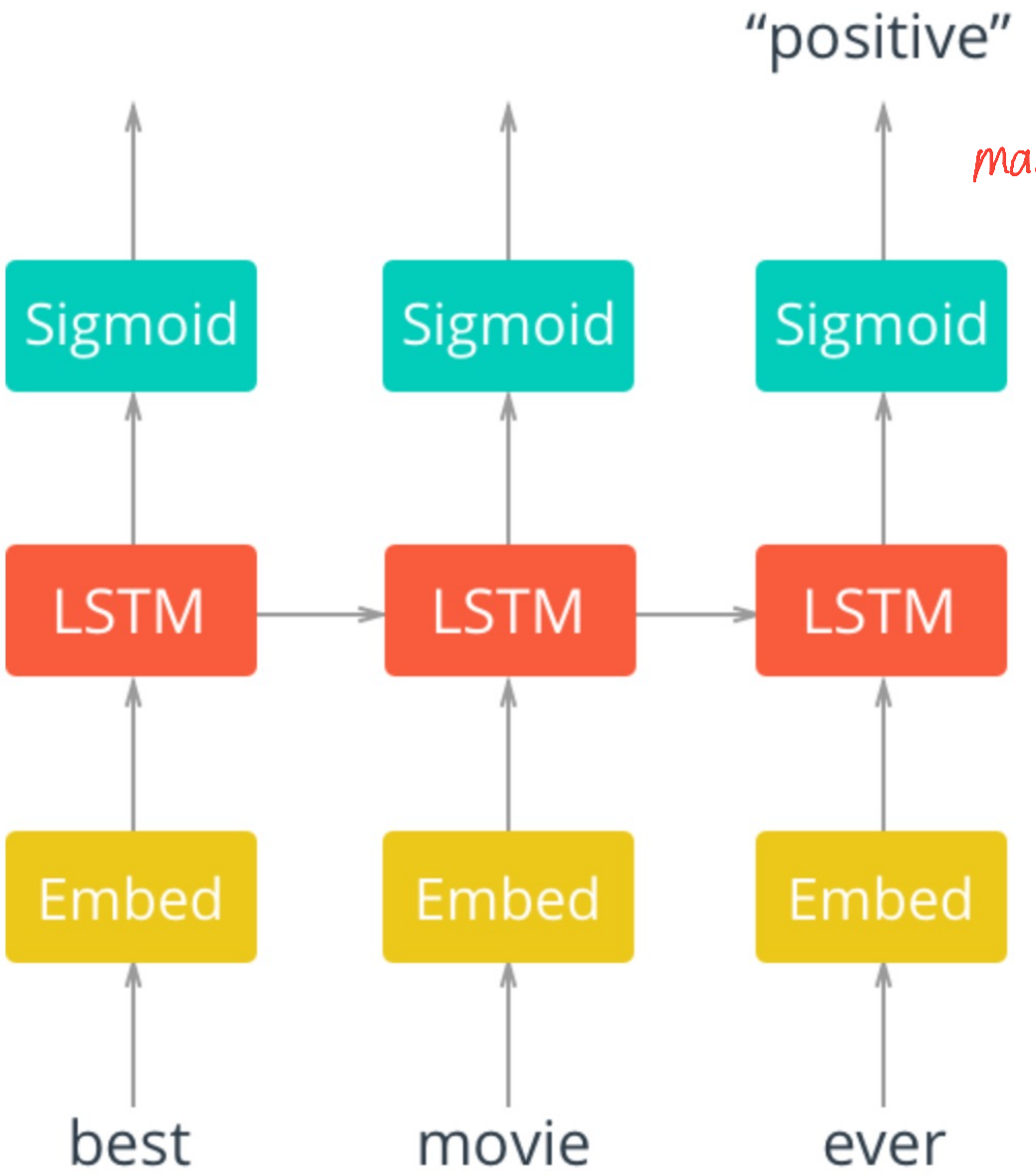
- Coronet has the best lines of all day cruisers.
- Bertram has a deep V hull and runs easily through seas.
- Pastel-colored 1980s day cruisers from Florida are ugly.
- I dislike old cabin cruisers.

I do not dislike cabin cruisers. (negation handling)

Polarity:

classifier pos / neg

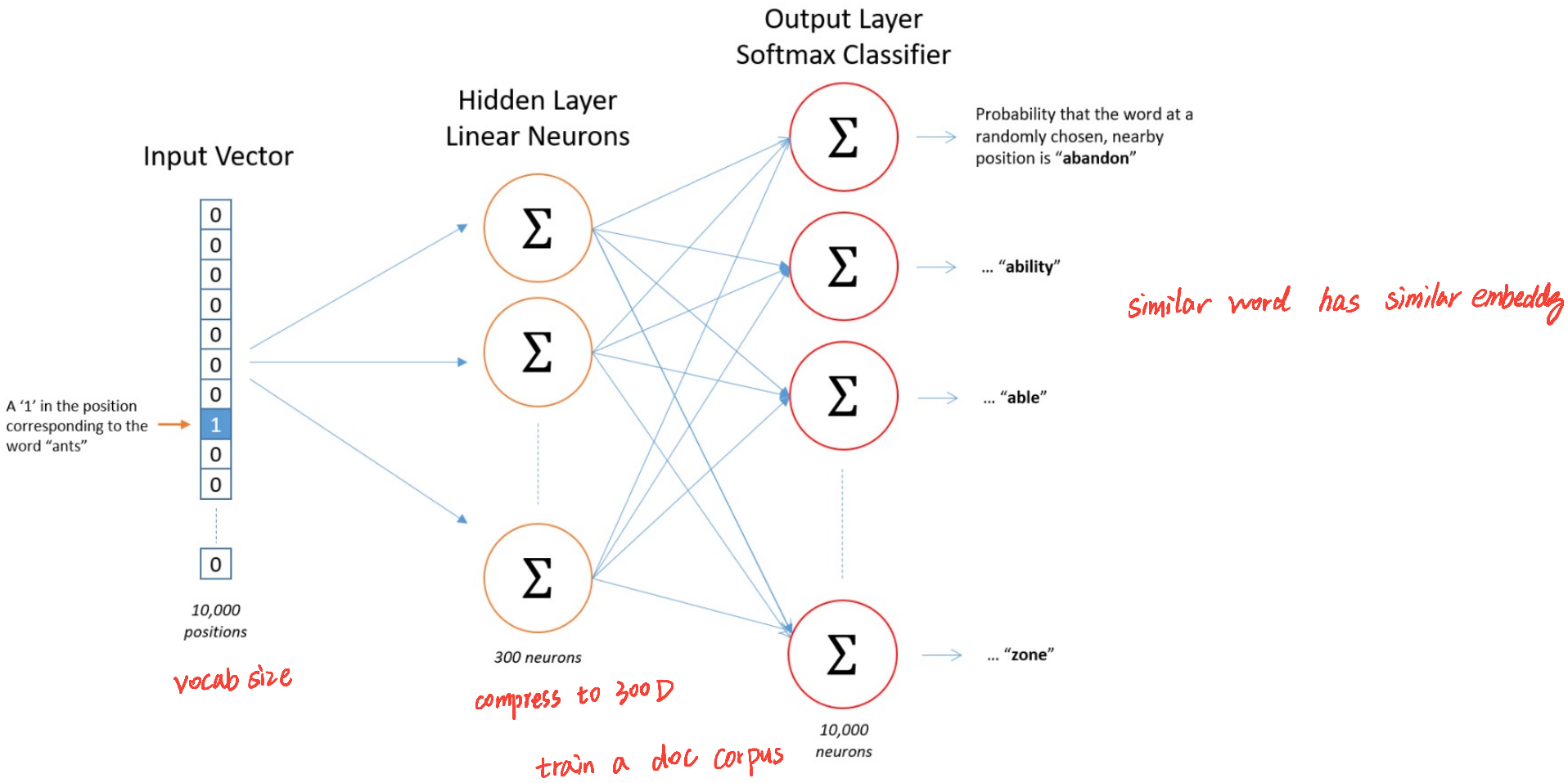
Beyond Polarity



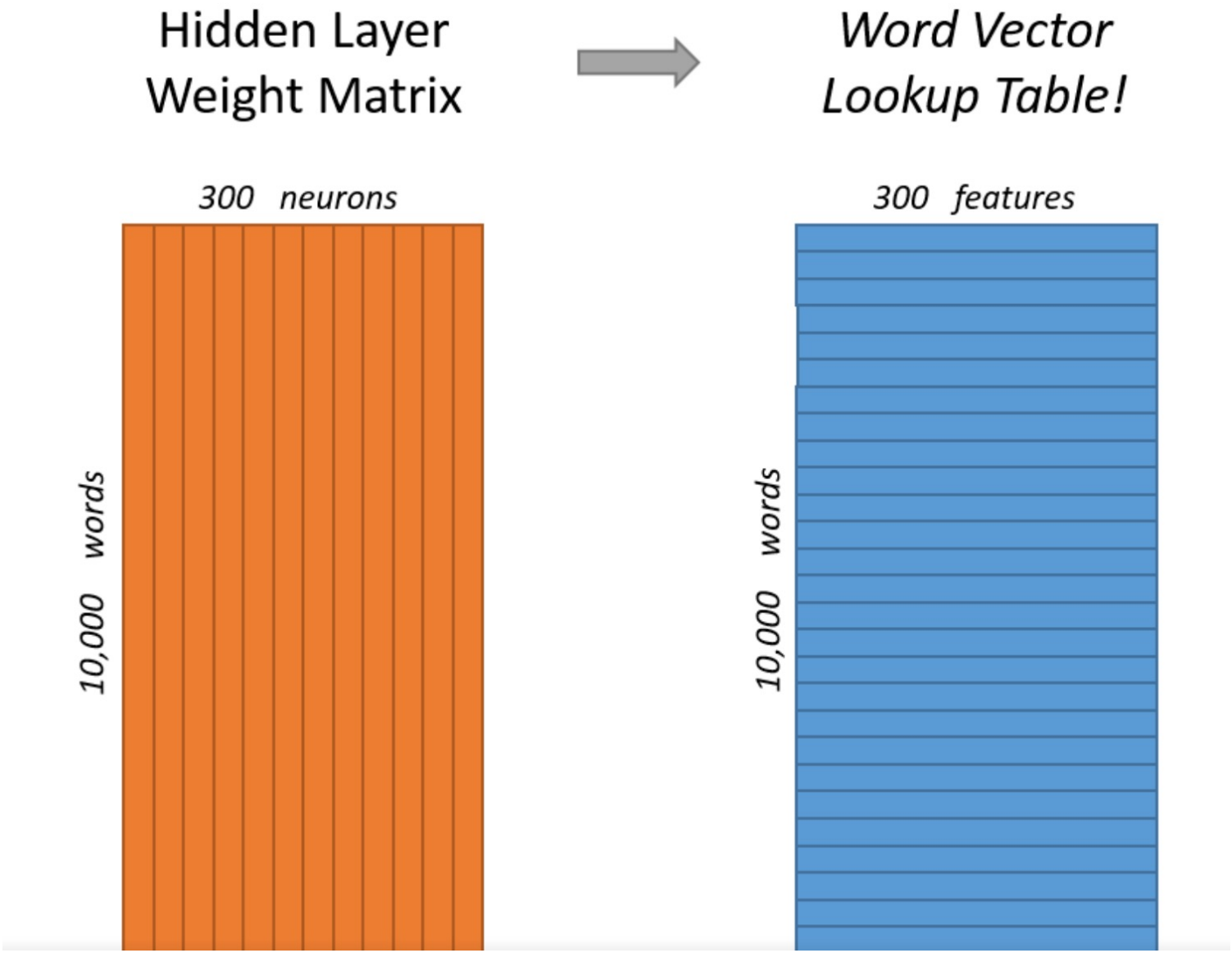
many to one seq model

Word vector Embedding

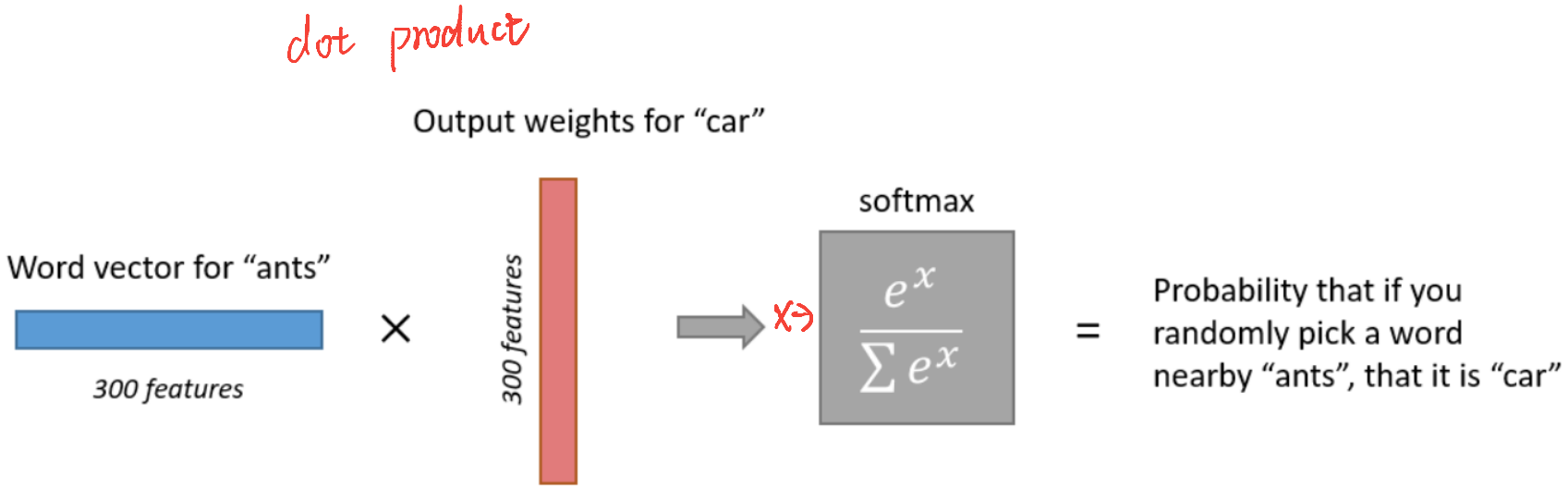
an unsupervised learning method



Word vectors

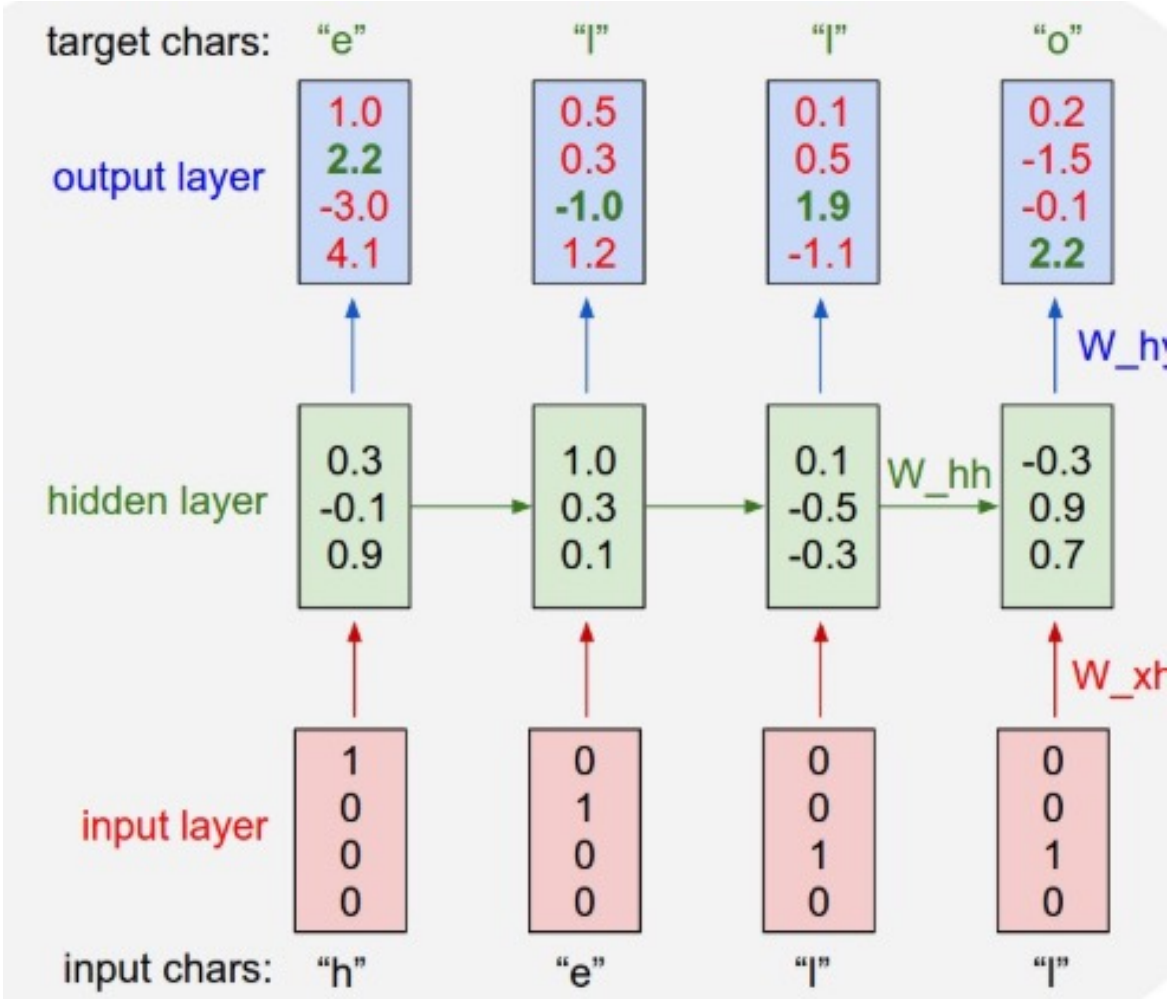


Output calculation



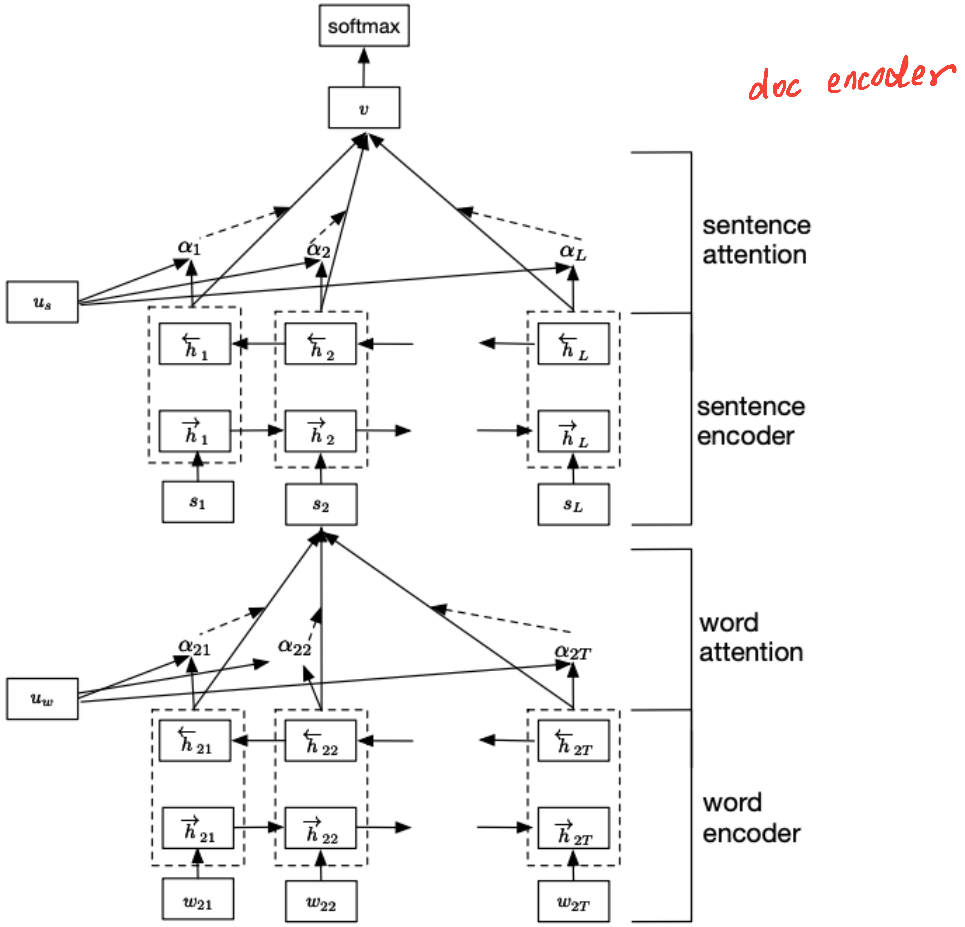
Training on negative samples improves this.

Character Level Language Model



Hierarchical Model

character → word → sentence → doc



Text Generation

Charisma / Power

January 2017

People who are powerful but uncharismatic will tend to be disliked. Their power makes them a target for criticism that they don't have the charisma to disarm. That was Hillary Clinton's problem. It also tends to be a problem for any CEO who is more of a builder than a schmoozer. And yet the builder-type CEO is (like Hillary) probably the best person for the job.

I don't think there is any solution to this problem. It's human nature. The best we can do is to recognize that it's happening, and to understand that being a magnet for criticism is sometimes a sign not that someone is the wrong person for a job, but that they're the right one.

Text generation databases: Paul Graham Essays, Project Gutenberg

RNN for Text Generation

2-layer LSTM

"The surprised in investors weren't going to raise money. I'm not the company with the time there are all interesting quickly, don't have to get off the same programmers. There's a super-angel round fundraising, why do you can do. If you have a different physical investment are become in people who reduced in a startup with the way to argument the acquirer could see them just that you're also the founders will part of users' affords that and an alternation to the idea. [2] Don't work at first member to see the way kids will seem in advance of a bad successful startup. And if you have to act the big company too."

train LSTM to complete sentences

Depth helps!

- 3-layer LSTM

```
PANDARUS:
Alas, I think he shall be come approached and the day
When little srain would be attain'd into being never fed,
And who is but a chain and subjects of his death,
I should not sleep.

Second Senator:
They are away this miseries, produced upon my soul,
Breaking and strongly should be buried, when I perish
The earth and thoughts of many states.

DUKE VINCENTIO:
Well, your wit is in the care of side and that.

Second Lord:
They would be ruled after this chamber, and
my fair nues begun out of the fact, to be conveyed,
Whose noble souls I'll have the heart of the wars.

Clown:
Come, sir, I will make did behold your worship.

VIOLA:
I'll drink it.
```

generate

Baby Names

Rudi Levette Berice Lussa Hany Mareanne Chrestina Carissy Marylen Hammine Janye Marlise Jacacrie
Hendred Romand Charienna Nenotto Ette Dorane Wallen Marly Darine Salina Elvyn Ersia Maralena Minoria Ellia
Charmin Antley Nerille Chelon Walmor Evena Jeryly Stachon Charisa Allisa Anatha Cathanie Geetra Alexie Jerin
Cassen Herbett Cossie Velen Daurenge Robester Shermond Terisa Licia Roselen Ferine Jayn Lusine Charyanne
Sales Sanny Resa Wallon Martine Merus Jelen Candica Wallin Tel Rachene Tarine Ozila Ketia Shanne Arnande
Karella Roselina Alessia Chasty Deland Berther Geamar Jackein Mellisand Sagdy Nenc Lessie Rasemy Guen
Gavi Milea Anneda Margoris Janin Rodelin Zeanna Elyne Janah Ferzina Susta Pey Castina

New names that have not been sampled before!

Evolution of a language model: Iteration 100

```
tyntd-iafhatawiaoihrdemot  lytdws  e ,tfti, astai f ogoh eoase rrranbyne 'nhthnee e
plia tklrgd t o idoe ns,smtt   h ne etie h,hregtrs nigtike,aoaenns lng
```

What did the network learn so far?

Words and spaces.

300 Iterations

```
"Tmont thithey" fomesscerliund
Keushey. Thom here
sheulke, anmerenith ol sivh I lalterthend Bleipile shuwy fil on aseterlome
coaniogennc Phe lism thond hon at. MeiDimorotion in ther thize."
```

Punctuation

Iteration 500

```
we counter. He stutn co des. His stanted out one ofler that concossions and was
to gearang reay Jotrets and with fre colt otf paitt thin wall. Which das stimn
```

Spelling short words

Iteration 700

```
Aftair fall unsuch that the hall for Prince Velzonski's that me of
her hearly, and behs to so arwage fiving were to it beloge, pavu say falling misfort
how, and Gogition is so overelical and ofter.
```

Word ordering

Image Captioning

. Examples of attending to the correct object (*white* indicates the attended regions, *underlines* indicated the corresponding word)

A woman is throwing a frisbee in a park.

A dog is standing on a hardwood floor.

A stop sign is on a road with a mountain in the background.

A little girl sitting on a bed with a teddy bear.

A group of people sitting on a boat in the water.

A giraffe standing in a forest with trees in the background.

Reading list

- Chapter 10 Goodfellow
- Miklov et al. Efficient Estimation of Word Representations in Vector Space ,2013

attention

- Luong et al. Effective Approaches to Attention-based Neural Machine Translation 2015
- https://arxiv.org/pdf/1409.0473.pdf

LSTM

- Hochreiter & Schmidhuber Long Sort Term Memory, 1997
- https://medium.com/datadriveninvestor/attention-in-rnns-321fbcd64f05
- http://colah.github.io/posts/2015-08-Understanding-LSTMs/
- https://lena-voita.github.io/nlp_course/seq2seq_and_attention.html