Chapter 8.1 - 8.4 incl.

May, 2nd 2022
Tassilo Henninger

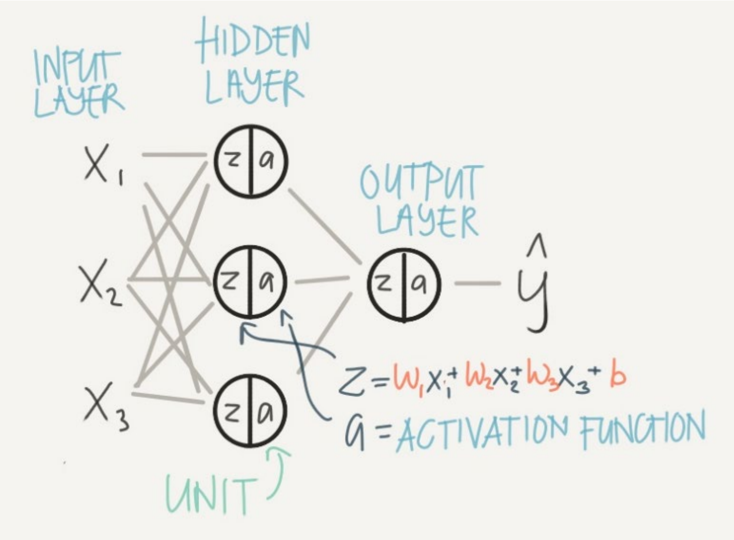
BHT Berliner Hochschule für Technik

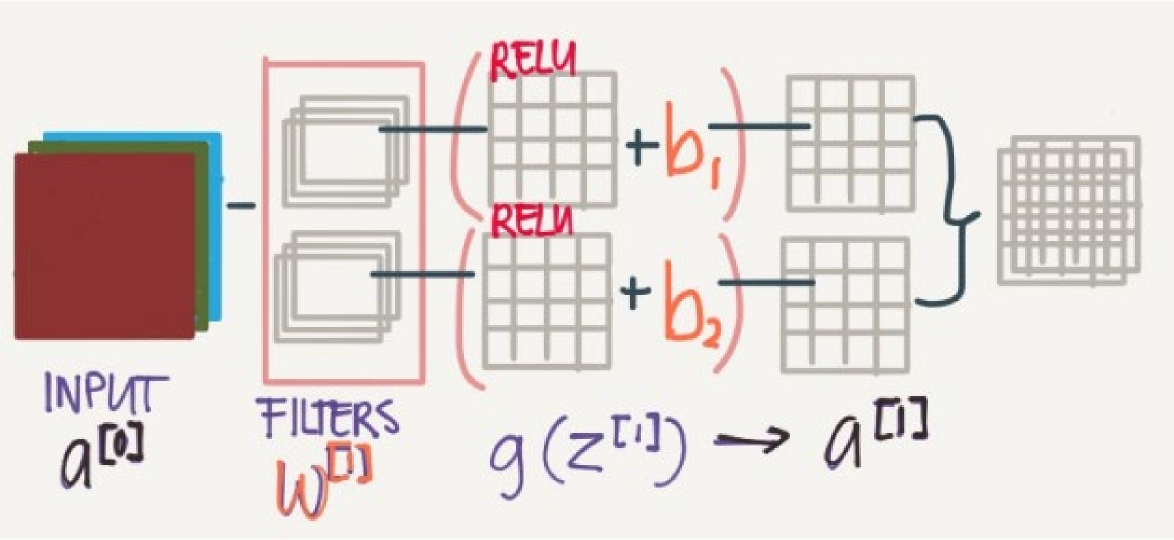
Studiere Zukunft

Agenda:

1. sequence models and sequence data
2. text preprocessing
3. language model (as the inspiration for the design of RNNs)
4. RNN

What we had so far…

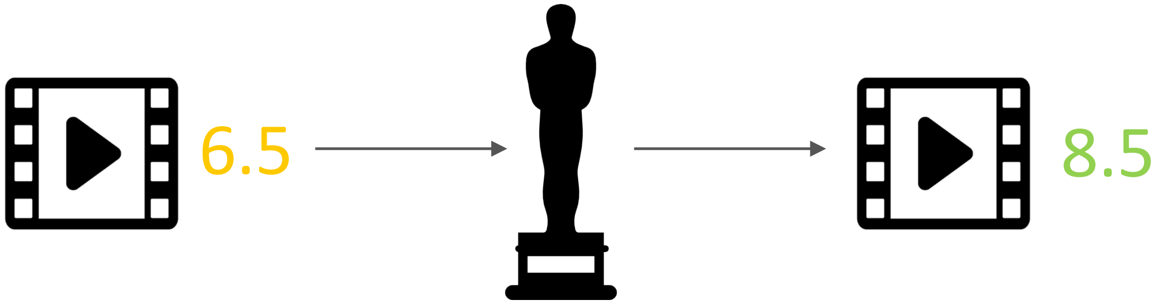




How can we handle sequence data?

How can „sequence data” look like?

data is not stationary:

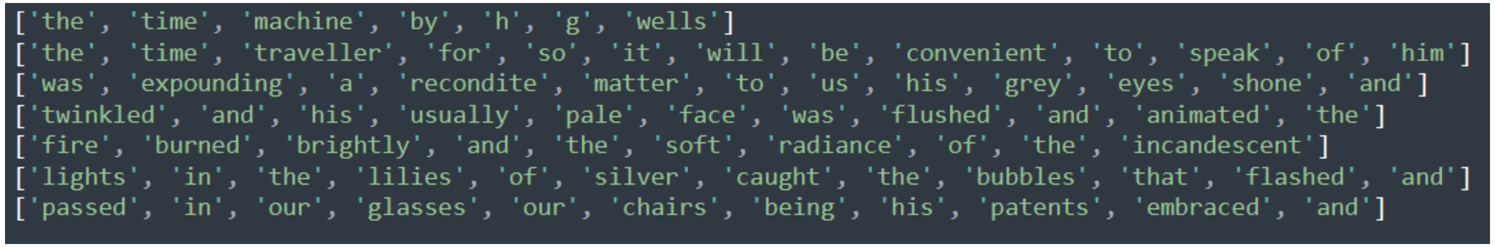


sequence matters:

- Music, speech, text, and videos are all sequential in nature
- ***”dog bites man”*** vs ***“man bites dog”***
- stock prices (*interpolation vs extrapolation*)

Prediction challenges with „sequence data”:

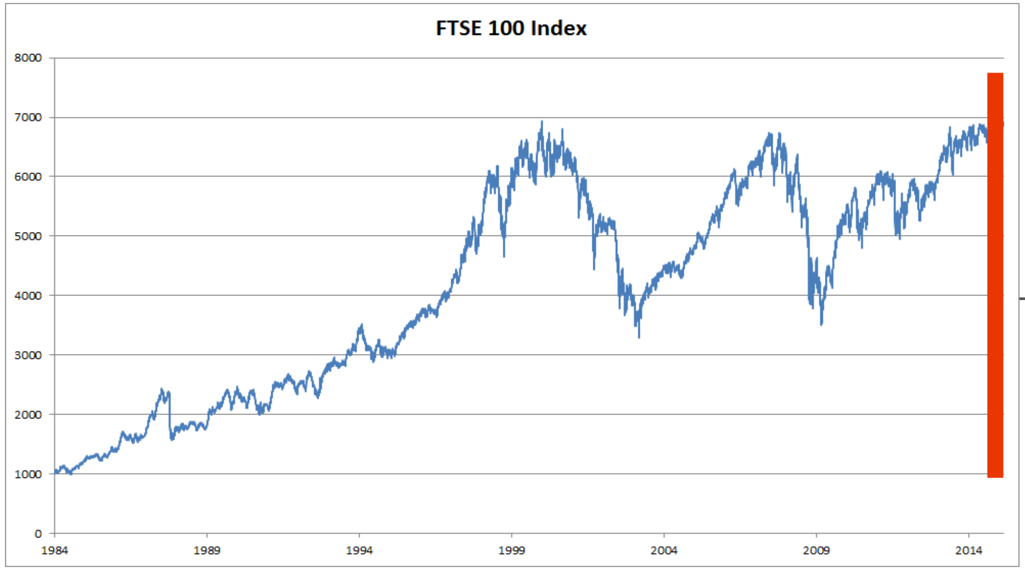
variate input length



```
['the', 'time', 'machine', 'by', 'h', 'g', 'wells']
['the', 'time', 'traveller', 'for', 'so', 'it', 'will', 'be', 'convenient', 'to', 'speak', 'of', 'him']
['was', 'expounding', 'a', 'recondite', 'matter', 'to', 'us', 'his', 'grey', 'eyes', 'shone', 'and']
['twinkled', 'and', 'his', 'usually', 'pale', 'face', 'was', 'flushed', 'and', 'animated', 'the']
['fire', 'burned', 'brightly', 'and', 'the', 'soft', 'radiance', 'of', 'the', 'incandescent']
['lights', 'in', 'the', 'lilies', 'of', 'silver', 'caught', 'the', 'bubbles', 'that', 'flashed', 'and']
['passed', 'in', 'our', 'glasses', 'our', 'chairs', 'being', 'his', 'patents', 'embraced', 'and']
```

→ ?

to continue the sequence



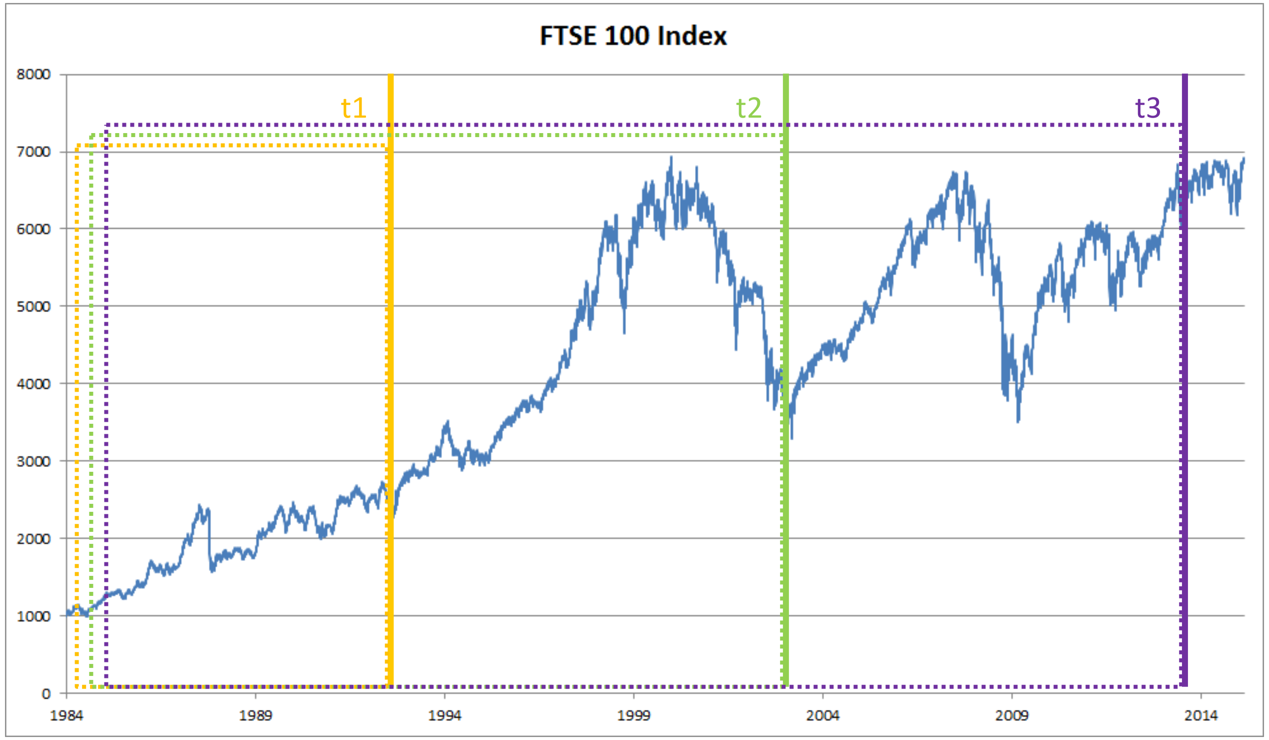
→ ?

Statistical Tools for overcoming those problems:

Autoregressive Models:

- assume that the potentially rather long sequence xt−1,...,x1 is not really necessary
- Use timespan of length σ and only use xt−1,...,xt− σ observations

→ input has always same length



Whenever this approximation is accurate, we say that the sequence satisfies a ***Markov condition***

$$P(x_{t+1} \mid x_t, \ldots, x_1) = P(x_{t+1} \mid x_t)$$

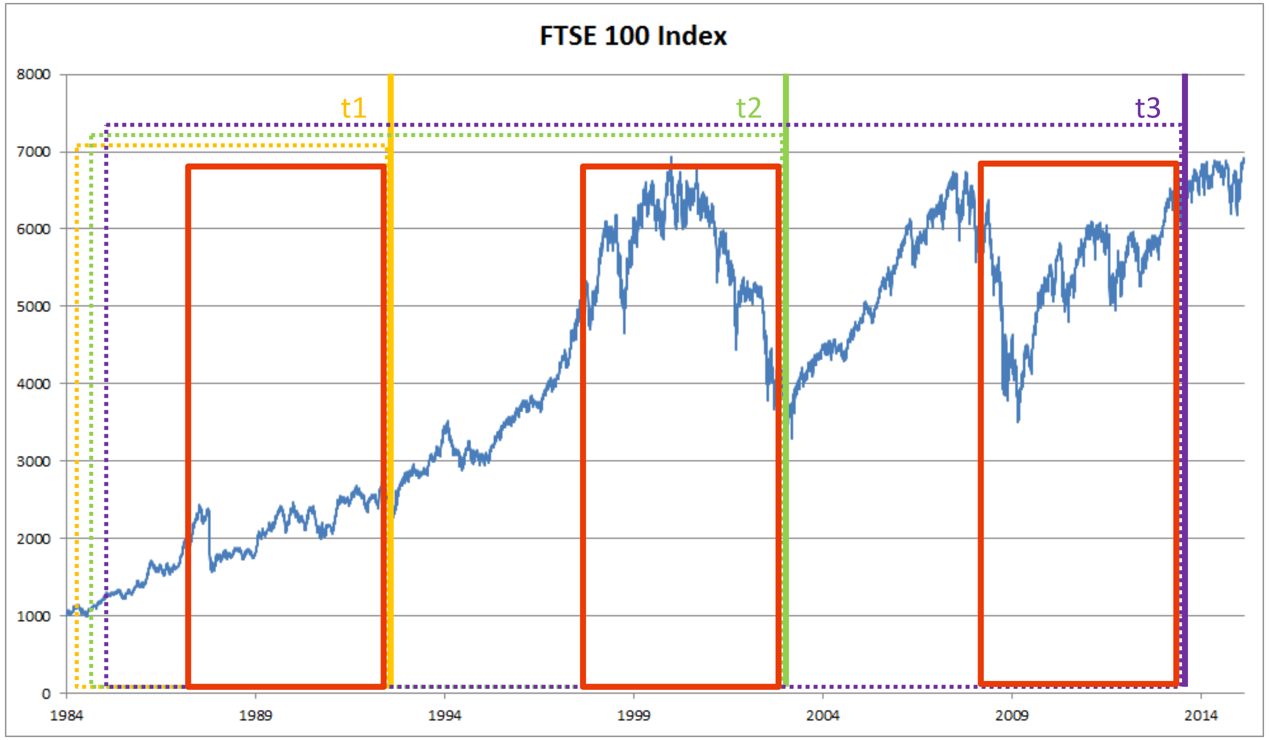
Statistical Tools for overcoming those problems:

Autoregressive Models:

- assume that the potentially rather long sequence xt−1,…,x1 is not really necessary

- Use timespan of length σ and only use xt−1,…,xt− σ observations

→ input has always same length



Whenever this approximation is accurate, we say that the sequence satisfies a ***Markov condition***

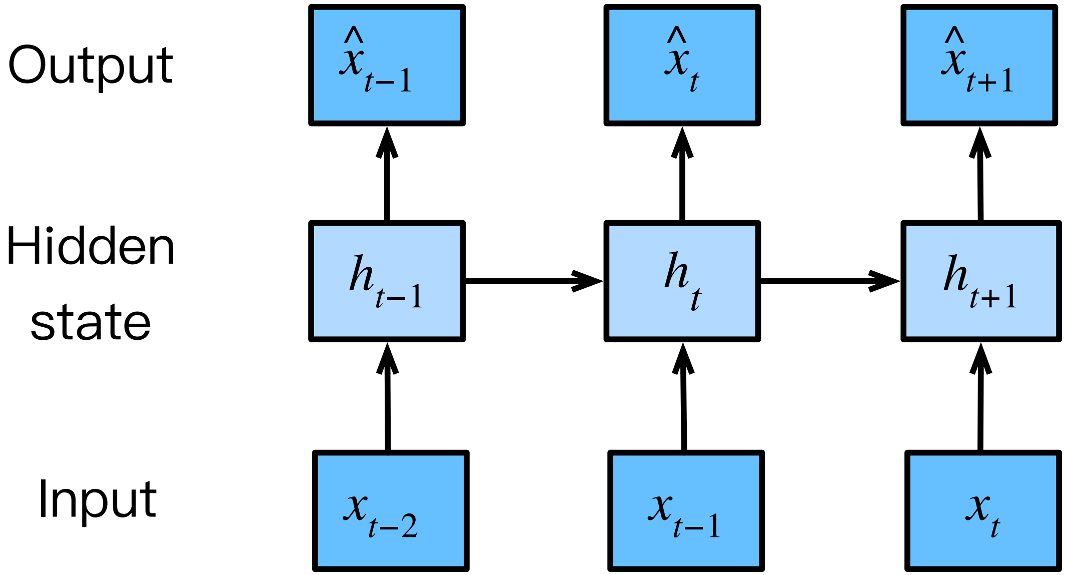
$$P(x_{t+1} \mid x_t, \ldots, x_1) = P(x_{t+1} \mid x_t)$$

Statistical Tools for overcoming those problems:

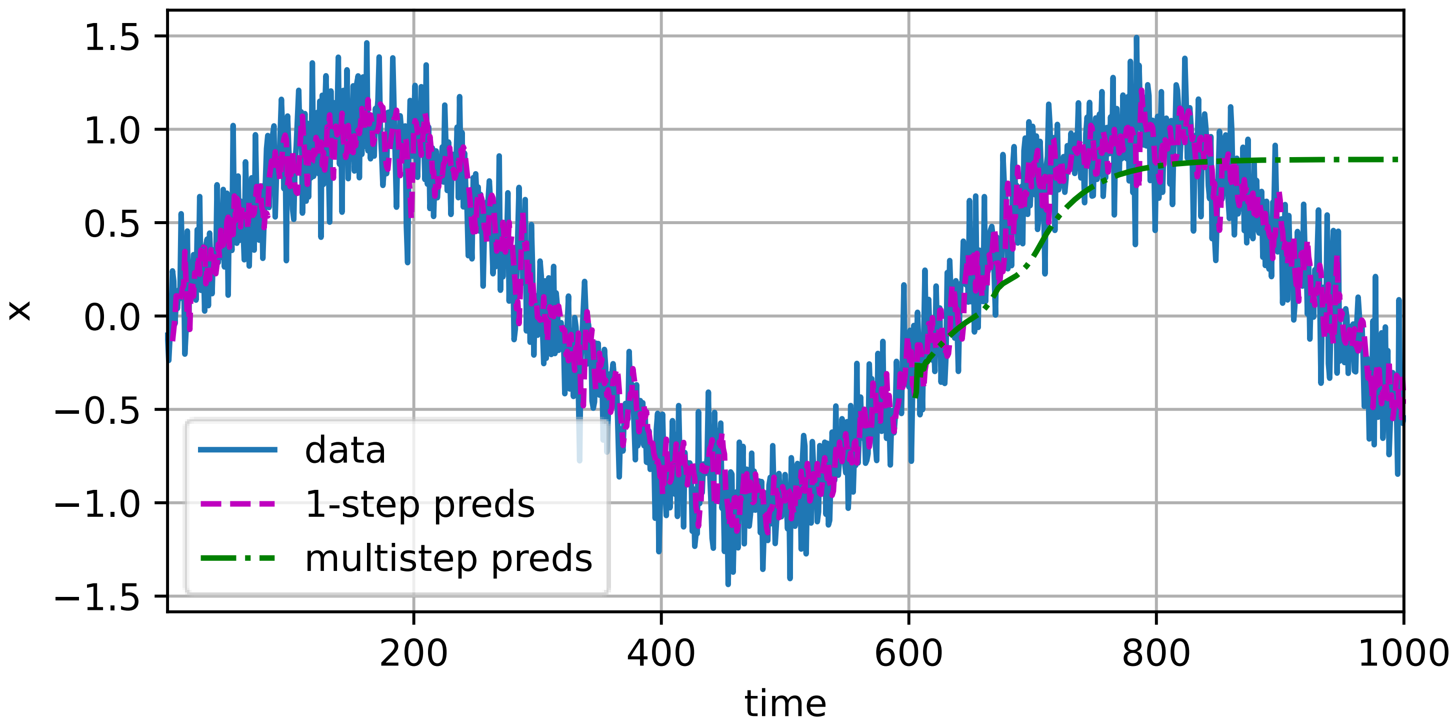
latent autoregressive models:

- The second strategy, is to keep some summary ht of the past observations

- Use summary in addition to the input for prediction



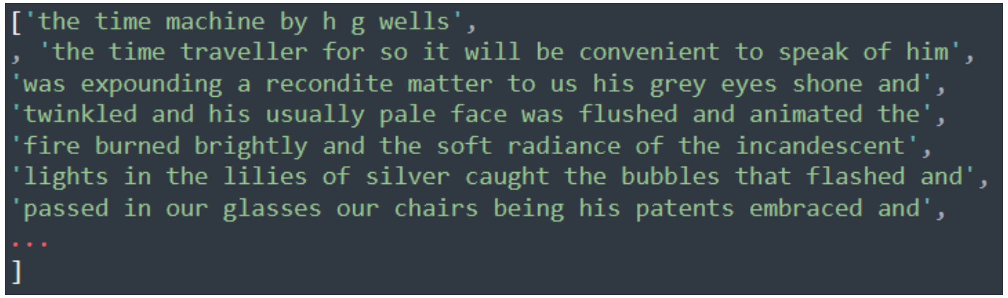
Concept in action!



$$
\begin{aligned}
\hat{x}_{605} &= f(x_{601}, x_{602}, x_{603}, x_{604}),\\
\hat{x}_{606} &= f(x_{602}, x_{603}, x_{604}, \hat{x}_{605}),\\
\hat{x}_{607} &= f(x_{603}, x_{604}, \hat{x}_{605}, \hat{x}_{606}),\\
\hat{x}_{608} &= f(x_{604}, \hat{x}_{605}, \hat{x}_{606}, \hat{x}_{607}),\\
\hat{x}_{609} &= f(\hat{x}_{605}, \hat{x}_{606}, \hat{x}_{607}, \hat{x}_{608}),\\
&\ldots
\end{aligned}
$$

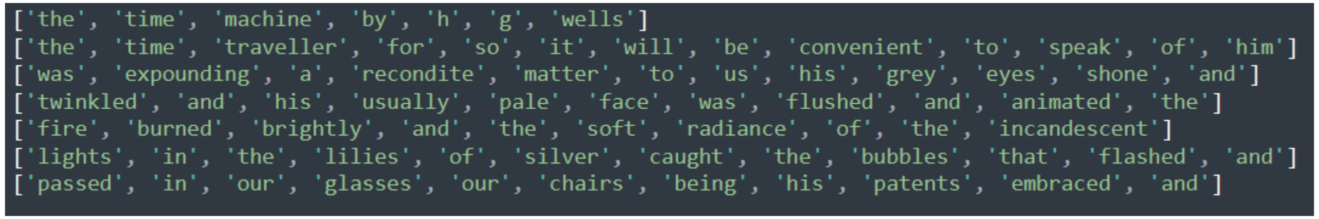
Text Preprocessing:

1. Load text as strings into memory.



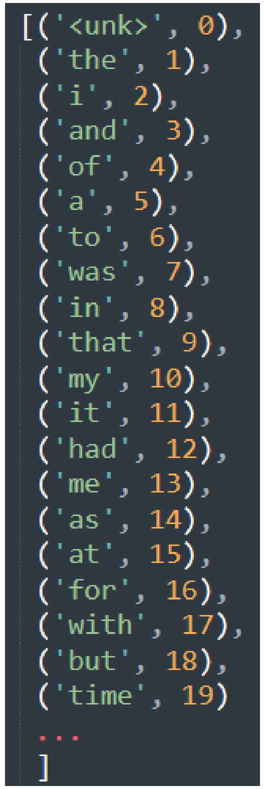
```
['the time machine by h g wells',
, 'the time traveller for so it will be convenient to speak of him',
'was expounding a recondite matter to us his grey eyes shone and',
'twinkled and his usually pale face was flushed and animated the',
'fire burned brightly and the soft radiance of the incandescent',
'lights in the lilies of silver caught the bubbles that flashed and',
'passed in our glasses our chairs being his patents embraced and',
...
]
```

2. Split strings into tokens



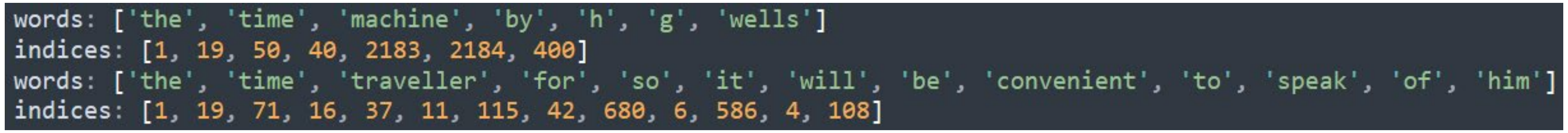
```
['the', 'time', 'machine', 'by', 'h', 'g', 'wells']
['the', 'time', 'traveller', 'for', 'so', 'it', 'will', 'be', 'convenient', 'to', 'speak', 'of', 'him']
['was', 'expounding', 'a', 'recondite', 'matter', 'to', 'us', 'his', 'grey', 'eyes', 'shone', 'and']
['twinkled', 'and', 'his', 'usually', 'pale', 'face', 'was', 'flushed', 'and', 'animated', 'the']
['fire', 'burned', 'brightly', 'and', 'the', 'soft', 'radiance', 'of', 'the', 'incandescent']
['lights', 'in', 'the', 'lilies', 'of', 'silver', 'caught', 'the', 'bubbles', 'that', 'flashed', 'and']
['passed', 'in', 'our', 'glasses', 'our', 'chairs', 'being', 'his', 'patents', 'embraced', 'and']
```

3. Build a vocabulary map

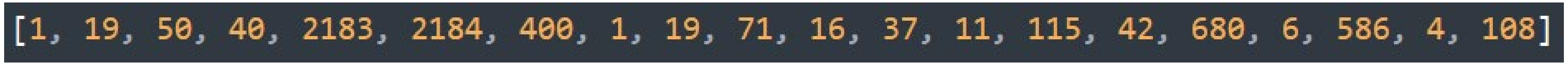


```
[('<unk>', 0),
 ('the', 1),
 ('i', 2),
 ('and', 3),
 ('of', 4),
 ('a', 5),
 ('to', 6),
 ('was', 7),
 ('in', 8),
 ('that', 9),
 ('my', 10),
 ('it', 11),
 ('had', 12),
 ('me', 13),
 ('as', 14),
 ('at', 15),
 ('for', 16),
 ('with', 17),
 ('but', 18),
 ('time', 19)
 ...
 ]
```

4. Convert text into sequences of numerical indices



```
words: ['the', 'time', 'machine', 'by', 'h', 'g', 'wells']
indices: [1, 19, 50, 40, 2183, 2184, 400]
words: ['the', 'time', 'traveller', 'for', 'so', 'it', 'will', 'be', 'convenient', 'to', 'speak', 'of', 'him']
indices: [1, 19, 71, 16, 37, 11, 115, 42, 680, 6, 586, 4, 108]
```



```
[1, 19, 50, 40, 2183, 2184, 400, 1, 19, 71, 16, 37, 11, 115, 42, 680, 6, 586, 4, 108]
```

Language Models and the Dataset:

goal of a *language model* is to estimate the joint probability of a sequence

$$P(x_1, x_2, \ldots, x_T).$$

['the', 'time', 'traveller', 'for', 'so', 'it', 'will', 'be', 'convenient', 'to', 'speak’]

ideal language model would be able to generate natural text just on its own

$$x_t \sim P(x_t \mid x_{t-1}, \ldots, x_1)$$

…'of', 'him'

language models are of great service even in their limited form

- In speech recognition or text summary: what is more likely

e.g. “to recognize speech” and “to wreck a nice beach”

How to calculate probability of a sequence?

$$P(x_1, x_2, \ldots, x_T) = \prod_{t=1}^{T} P(x_t \mid x_1, \ldots, x_{t-1}).$$

$$P(\text{deep}, \text{learning}, \text{is}, \text{fun}) = P(\text{deep})P(\text{learning} \mid \text{deep})P(\text{is} \mid \text{deep}, \text{learning})P(\text{fun} \mid \text{deep}, \text{learning}, \text{is})$$

Calculate probabilities based on relative word frequency:

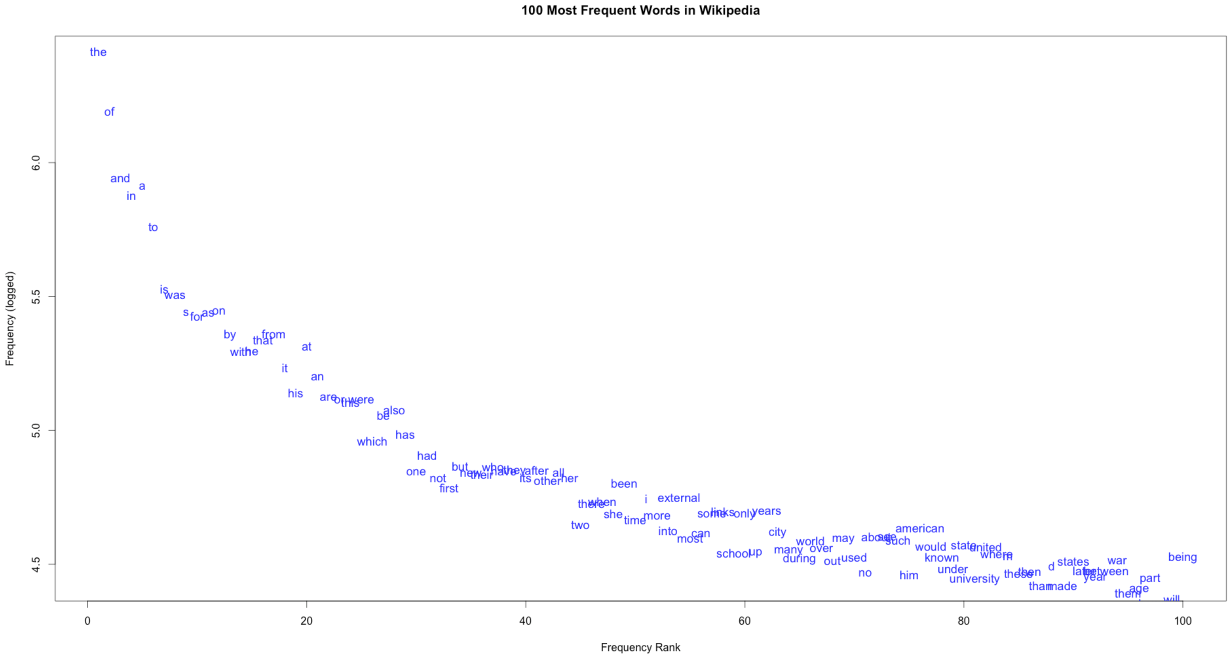
$$\hat{P}(\text{learning} \mid \text{deep}) = \frac{n(\text{deep, learning})}{n(\text{deep})},$$

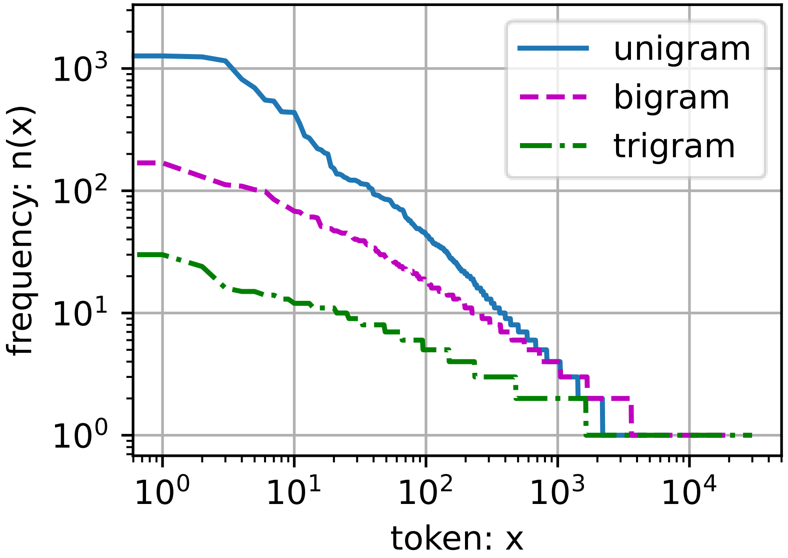
$$\begin{aligned} \hat{P}(x) & = \frac{n(x) + \epsilon_1/m}{n + \epsilon_1}, \\ \hat{P}(x' \mid x) & = \frac{n(x, x') + \epsilon_2 \hat{P}(x')}{n(x) + \epsilon_2}, \\ \hat{P}(x'' \mid x, x') & = \frac{n(x, x', x'') + \epsilon_3 \hat{P}(x'')}{n(x, x') + \epsilon_3}. \end{aligned}$$

This works fairly well, but only for frequent words…

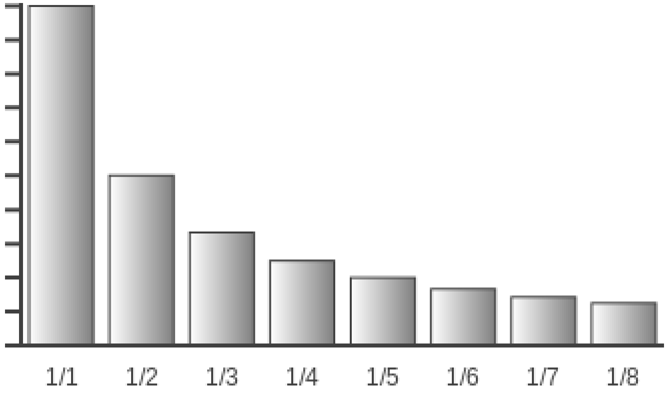
Possible solution: ***Laplace smoothing*** adds a small constant to all counts

Natural Language Statistics - Zipf's law:

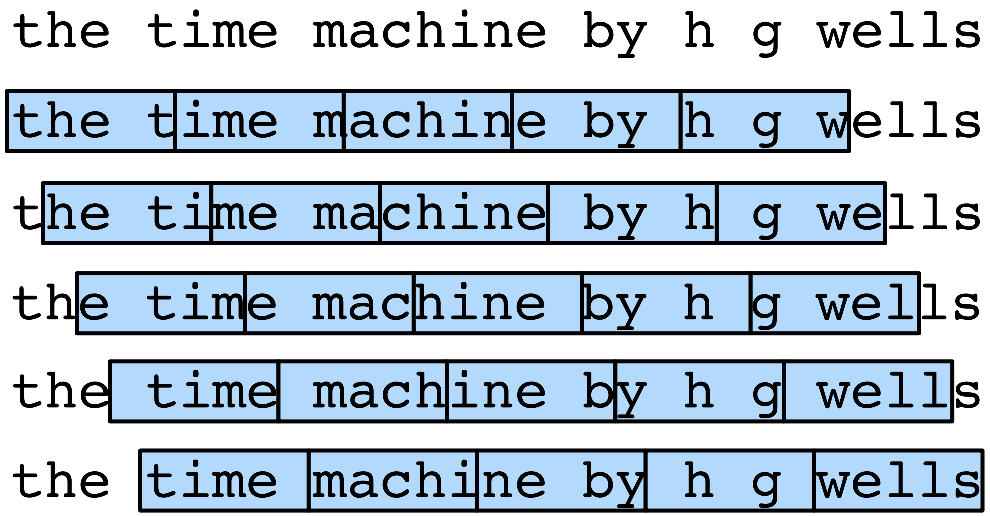




probability formulae that involve one, two, and three variables are typically referred to as *unigram*, *bigram*, and *trigram* models



How to process sequence data for NN?



the time machine by h g wells

the time machine by h g wells

the time machine by h g wells

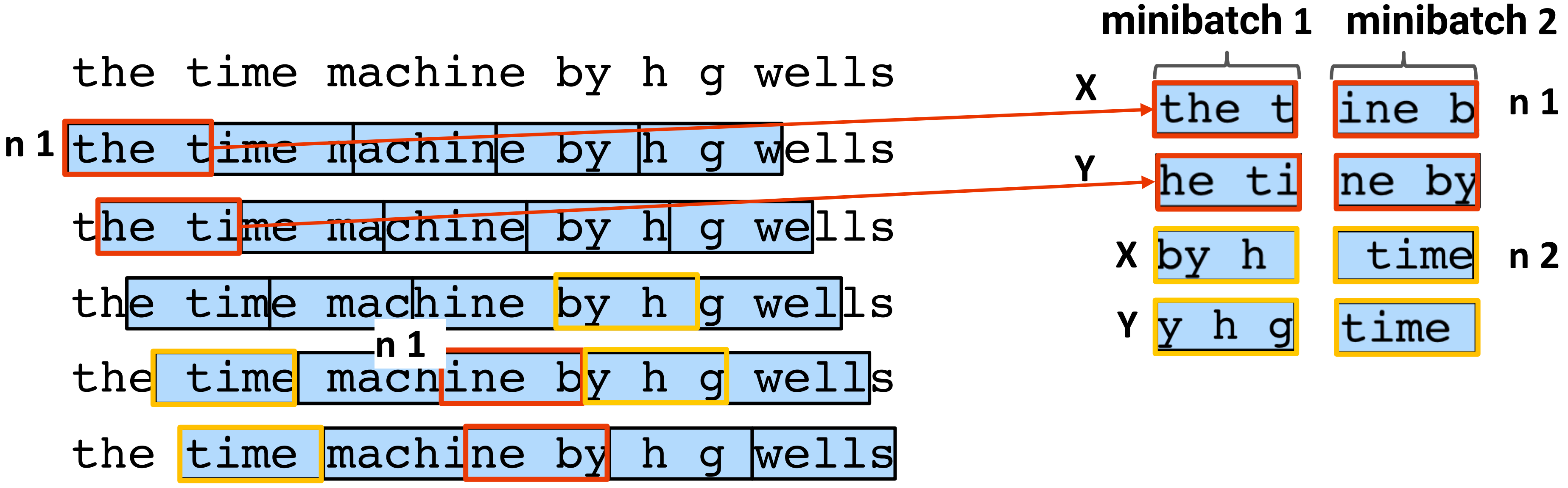
the time machine by h g wells

the time machine by h g wells

the time machine by h g wells

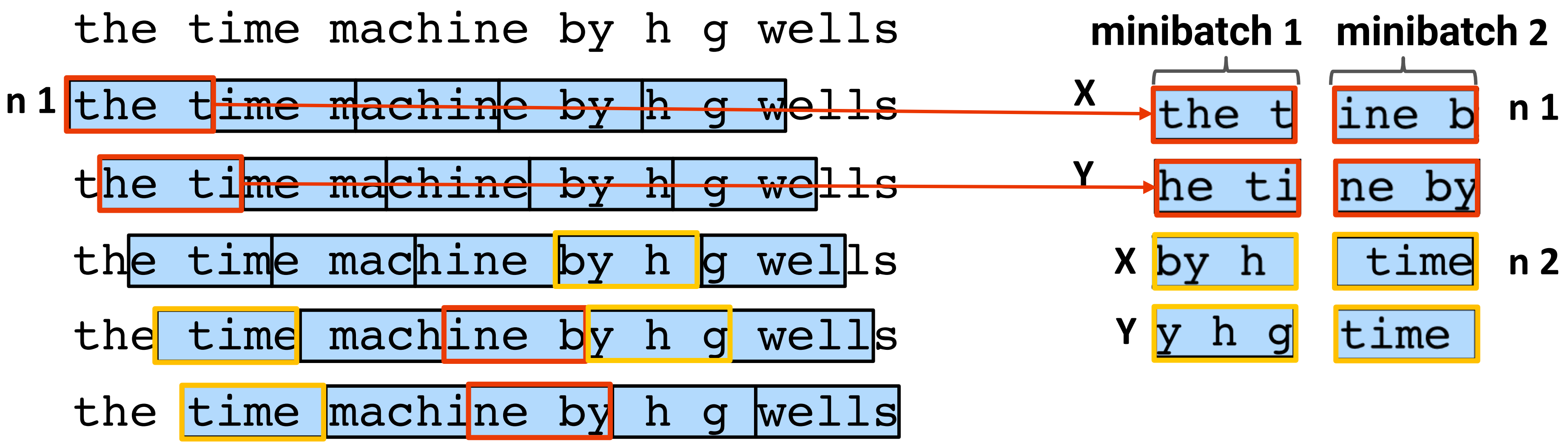
n = 5

How to process sequence data for NN?


minibatch 1
minibatch 2
the time machine by h g wells
n 1
the time machine by h g wells
the time machine by h g wells
the time machine by h g wells
n 1
the time machine by h g wells
the time machine by h g wells
X
the t
ine b
n 1
Y
he ti
ne by
X
by h
time
n 2
Y
y h g
time

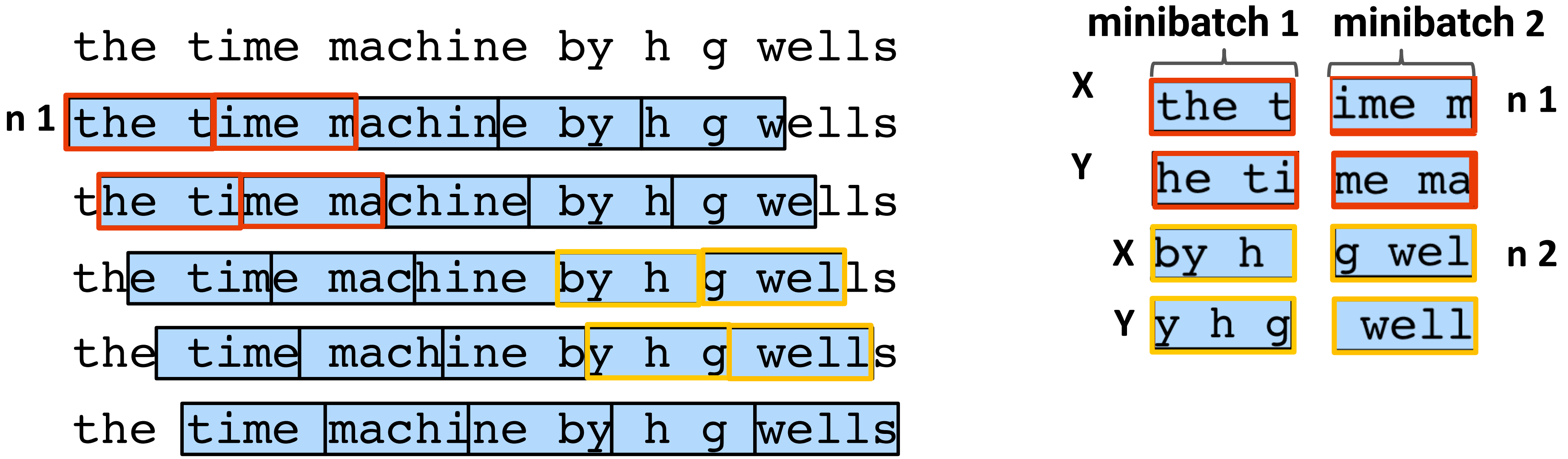
Random sampling

How to process sequence data for NN?



the time machine by h g wells
n 1
the time machine by h g wells
the time machine by h g wells
the time machine by h g wells
the time machine by h g wells
the time machine by h g wells
minibatch 1
minibatch 2
X
the t
ine b
n 1
Y
he ti
ne by
X
by h
time
n 2
Y
y h g
time

Random sampling

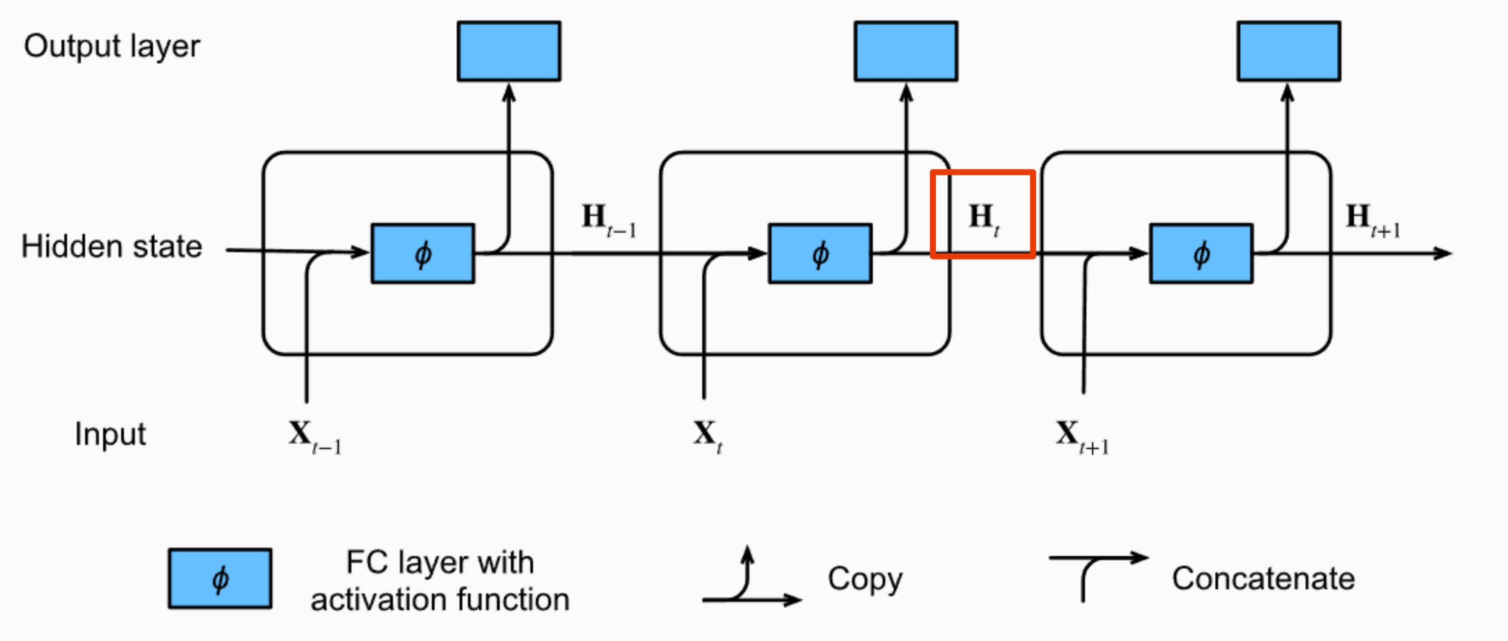


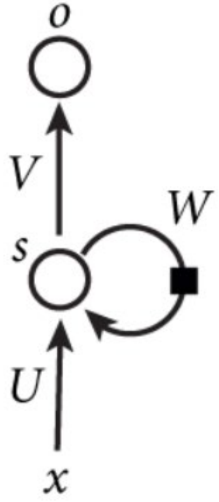
the time machine by h g wells
n 1
the time machine by h g wells
the time machine by h g wells
the time machine by h g wells
the time machine by h g wells
the time machine by h g wells
minibatch 1
minibatch 2
X
the t
ime m
n 1
Y
he ti
me ma
X
by h
g wel
n 2
Y
y h g
well

Sequential Partitioning

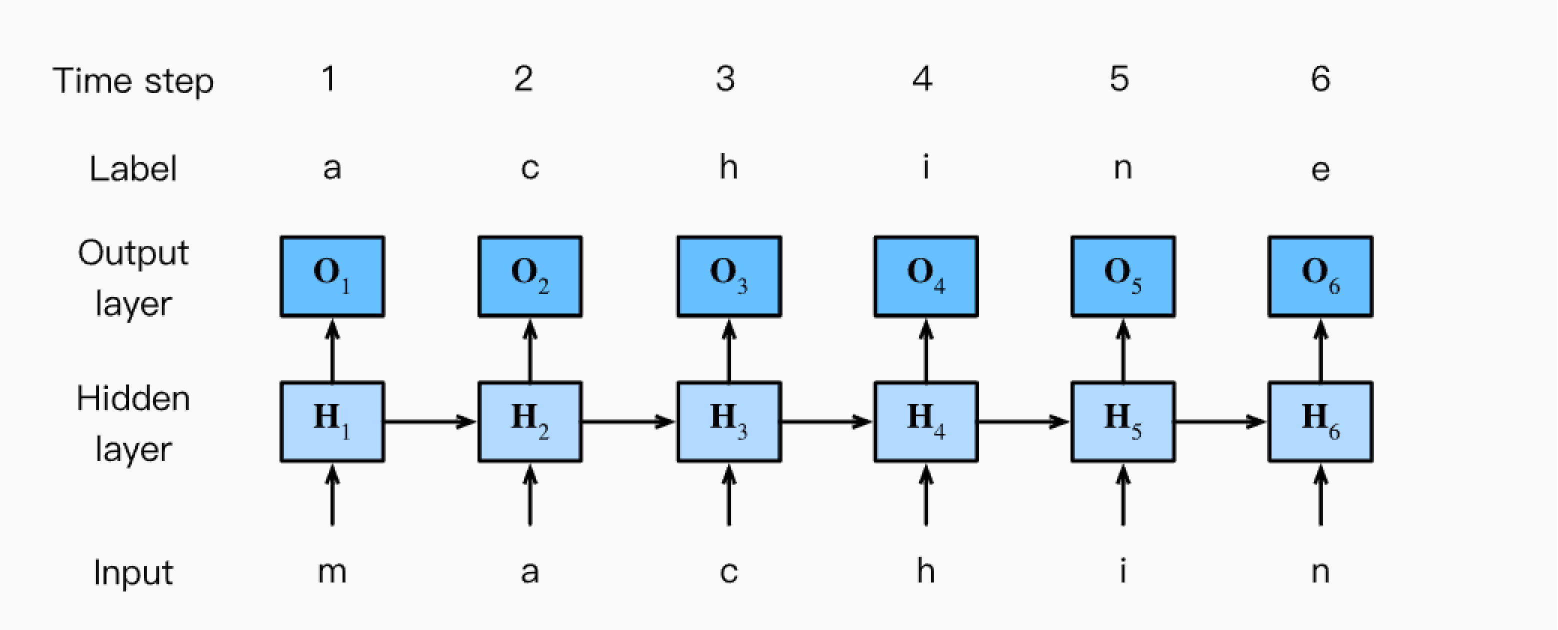
Recurrent Neural Networks:

hidden states: $$\mathbf{H}_t = \phi(\mathbf{X}_t \mathbf{W}_{xh} \boxed{+ \mathbf{H}_{t-1} \mathbf{W}_{hh}} + \mathbf{b}_h).$$



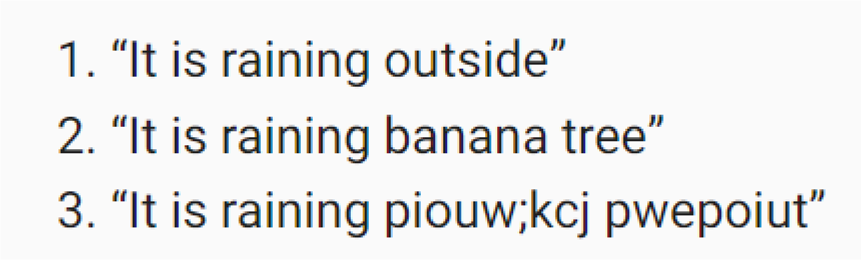


RNN-based Character-Level Language Models:

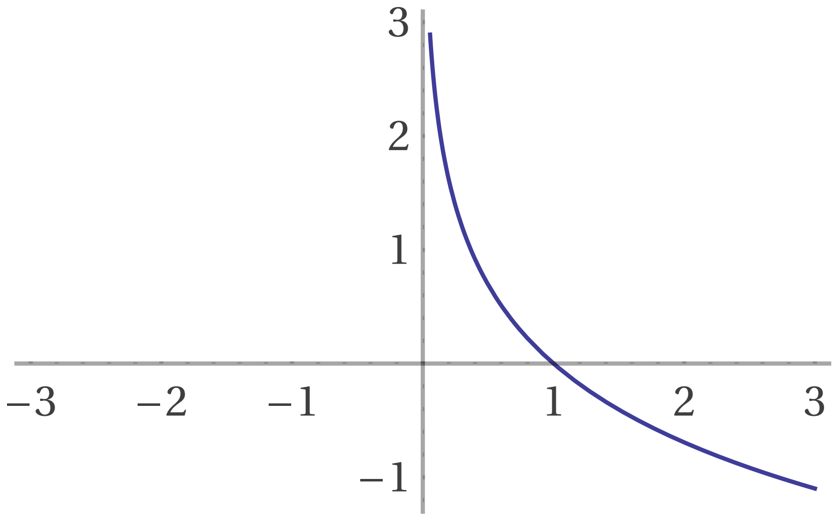


Perplexity and cross-entropy loss:

What is the best quality? How surprising is the text?



1. “It is raining outside”
2. “It is raining banana tree”
3. “It is raining piouw;kcj pwepoiut”



(*x* from −3 to 3)

Computed by Wolfram|Alpha

So we can measure it by the **cross-entropy loss** averaged over all the n tokens of a sequence.

$$\frac{1}{n}\sum_{t=1}^{n} -\log P(x_t \mid x_{t-1}, \ldots, x_1),$$

Or the **complexity**

$$\exp\left(-\frac{1}{n}\sum_{t=1}^{n} \log P(x_t \mid x_{t-1}, \ldots, x_1)\right)$$

Any Questions?