Foundations of Deep Learning

ALF

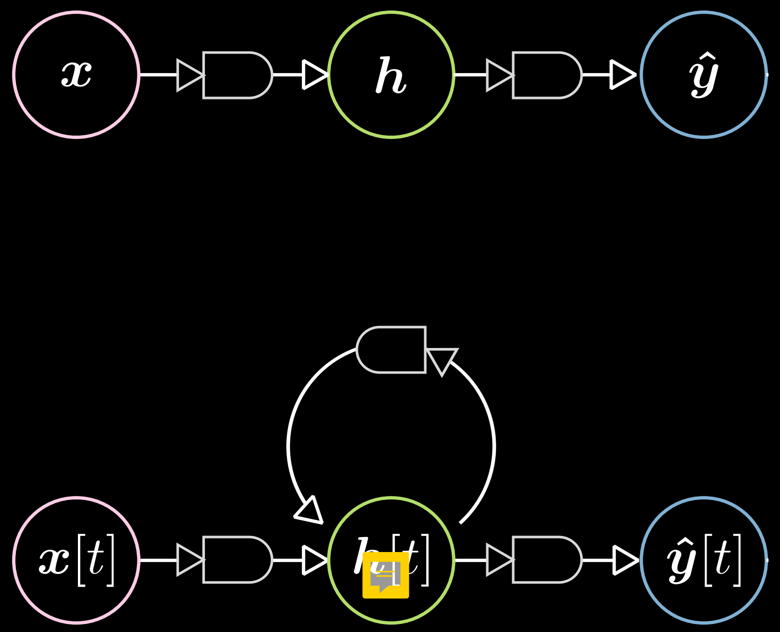
Alfredo Canziani

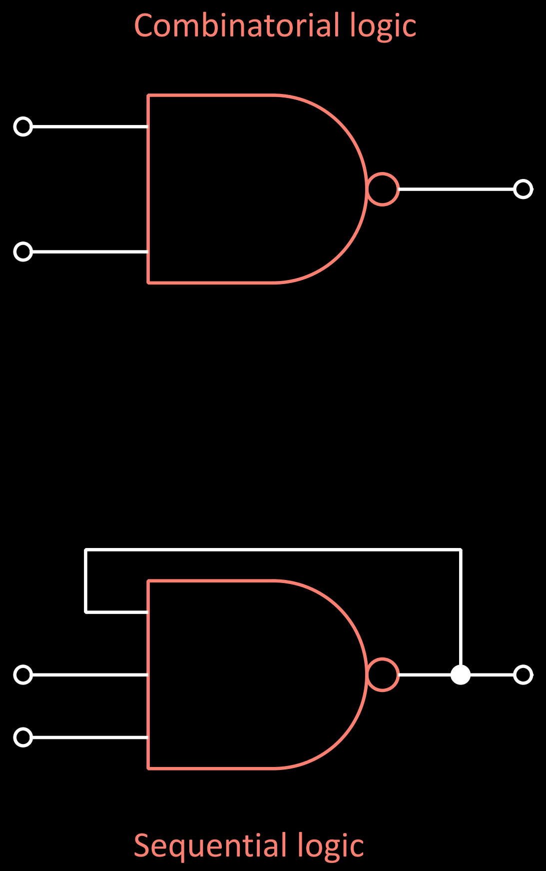
@alfcnz

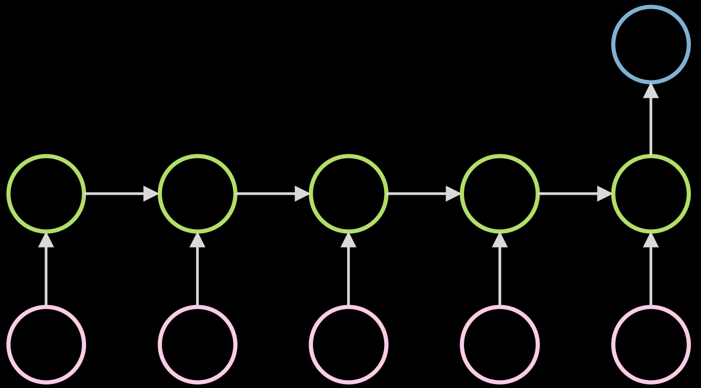
Recurrent Neural Nets

Handling sequential data

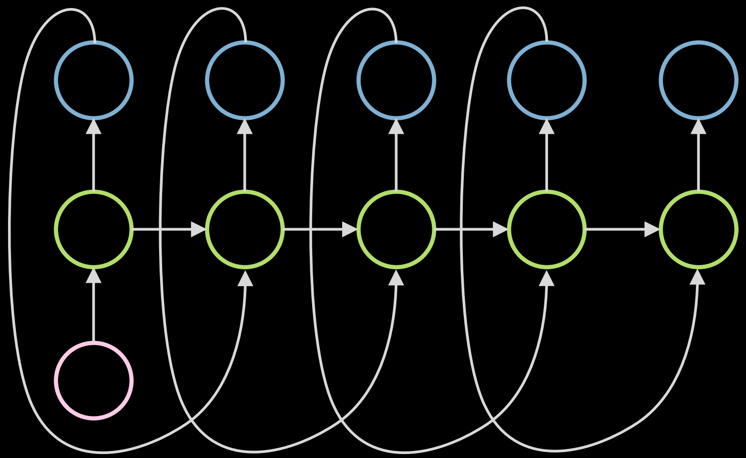
Vanilla and Recurrent NN



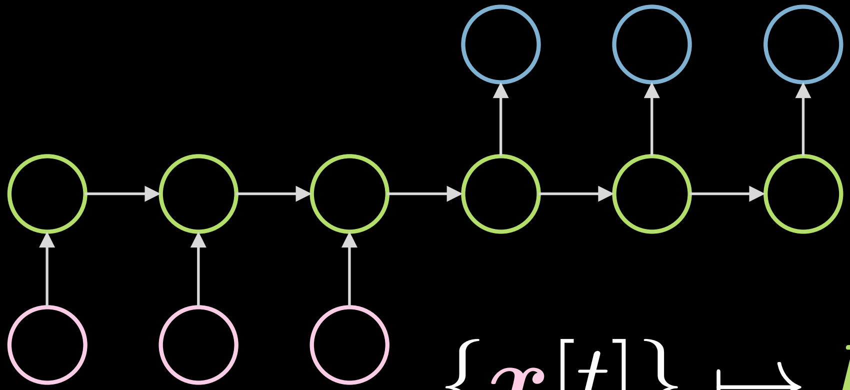




$\{\boldsymbol{x}[t]\} \mapsto \hat{\boldsymbol{y}}[T]$ seq $\mapsto$ vec

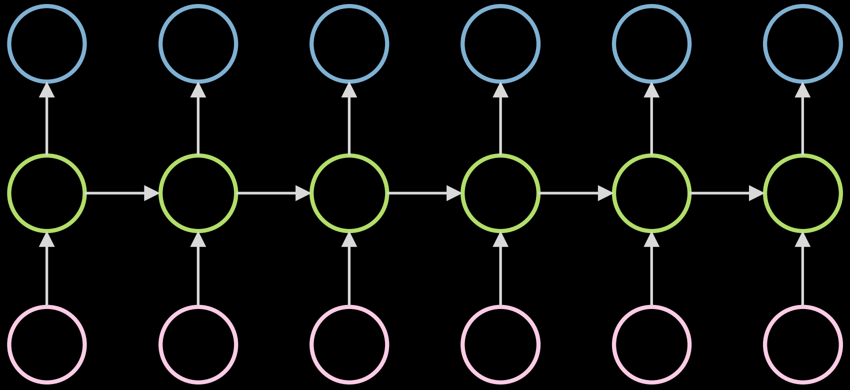


$\boldsymbol{x}[1] \mapsto \{\hat{\boldsymbol{y}}[t]\}$ vec $\mapsto$ seq



$\{\boldsymbol{x}[t]\} \mapsto \boldsymbol{h} \mapsto \{\hat{\boldsymbol{y}}[t]\}$

seq $\mapsto$ vec $\mapsto$ seq



$\{\boldsymbol{x}[t]\} \mapsto \{\hat{\boldsymbol{y}}[t]\}$ seq $\mapsto$ seq

Vinyals *et al.* (2016) Show and Tell: Lessons learned from the 2015 MSCOCO Image Captioning Challenge

Learning to execute

- **Input:**

```
j=8584
for x in range(8):
   j+=920
b=(1500+j)
print((b+7567))
```

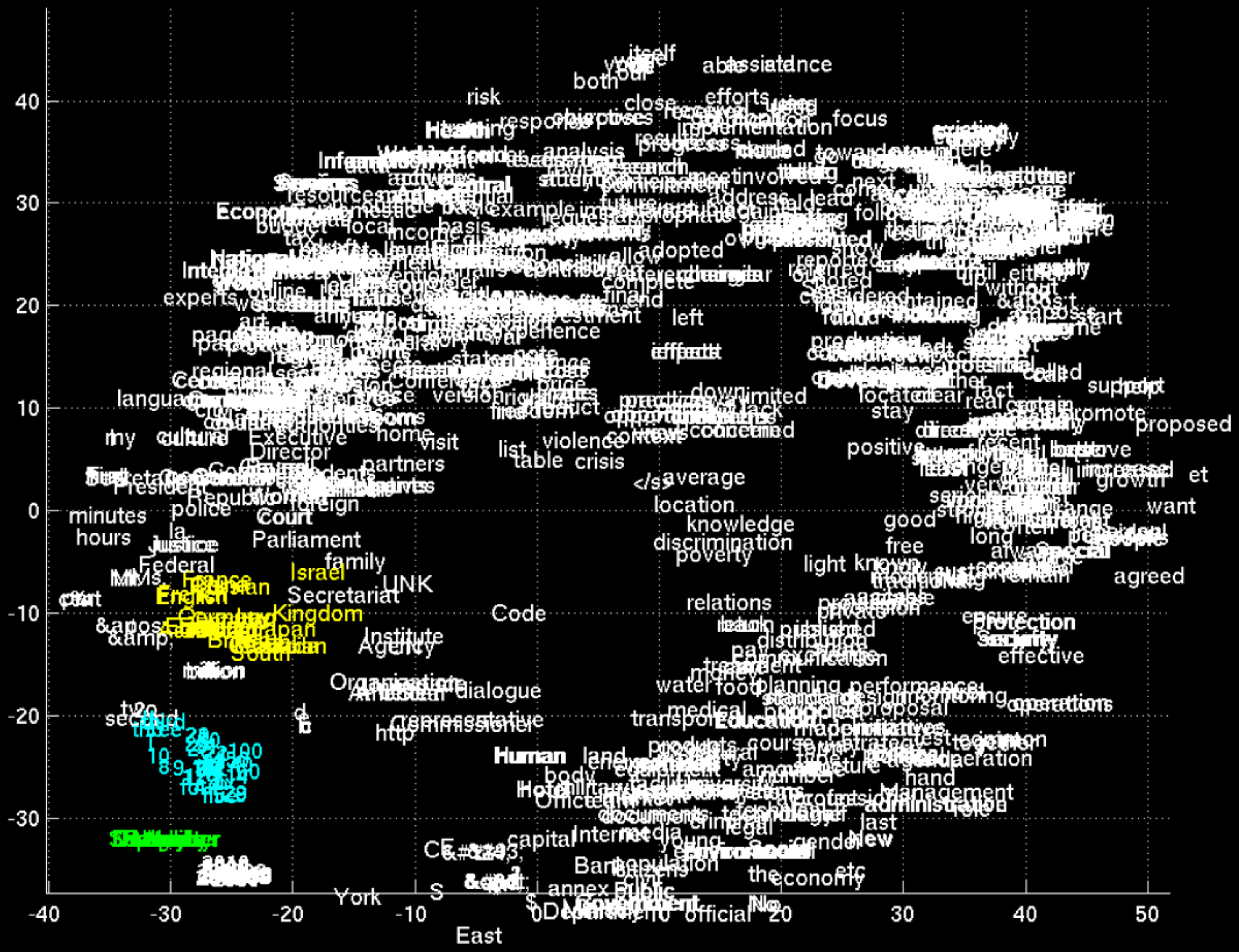
- Target: 25011.

- **Input:**

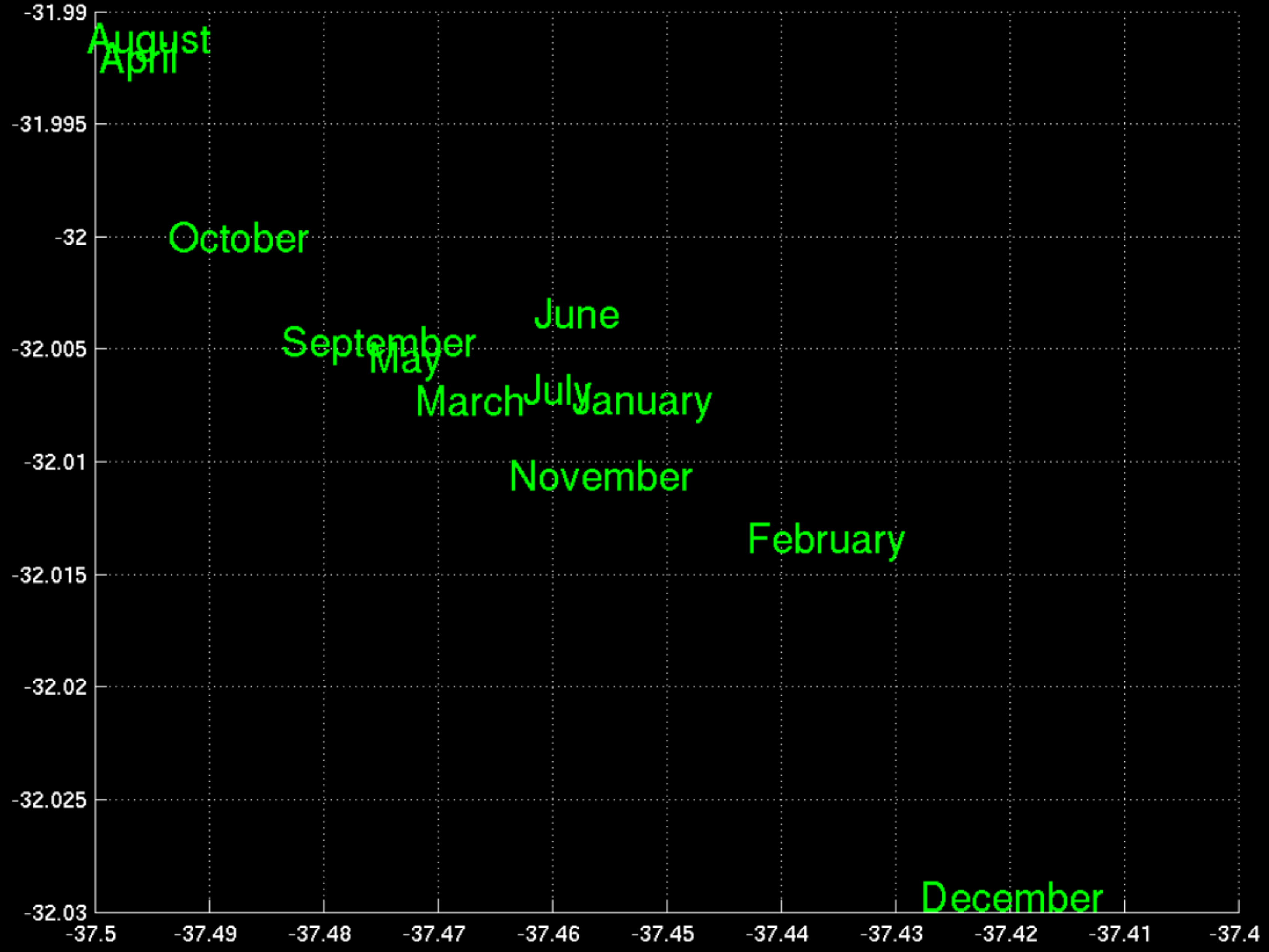
```
i=8827
c=(i-5347)
print((c+8704) if
2641<8500 else 5308)
```

- Target: 12184.

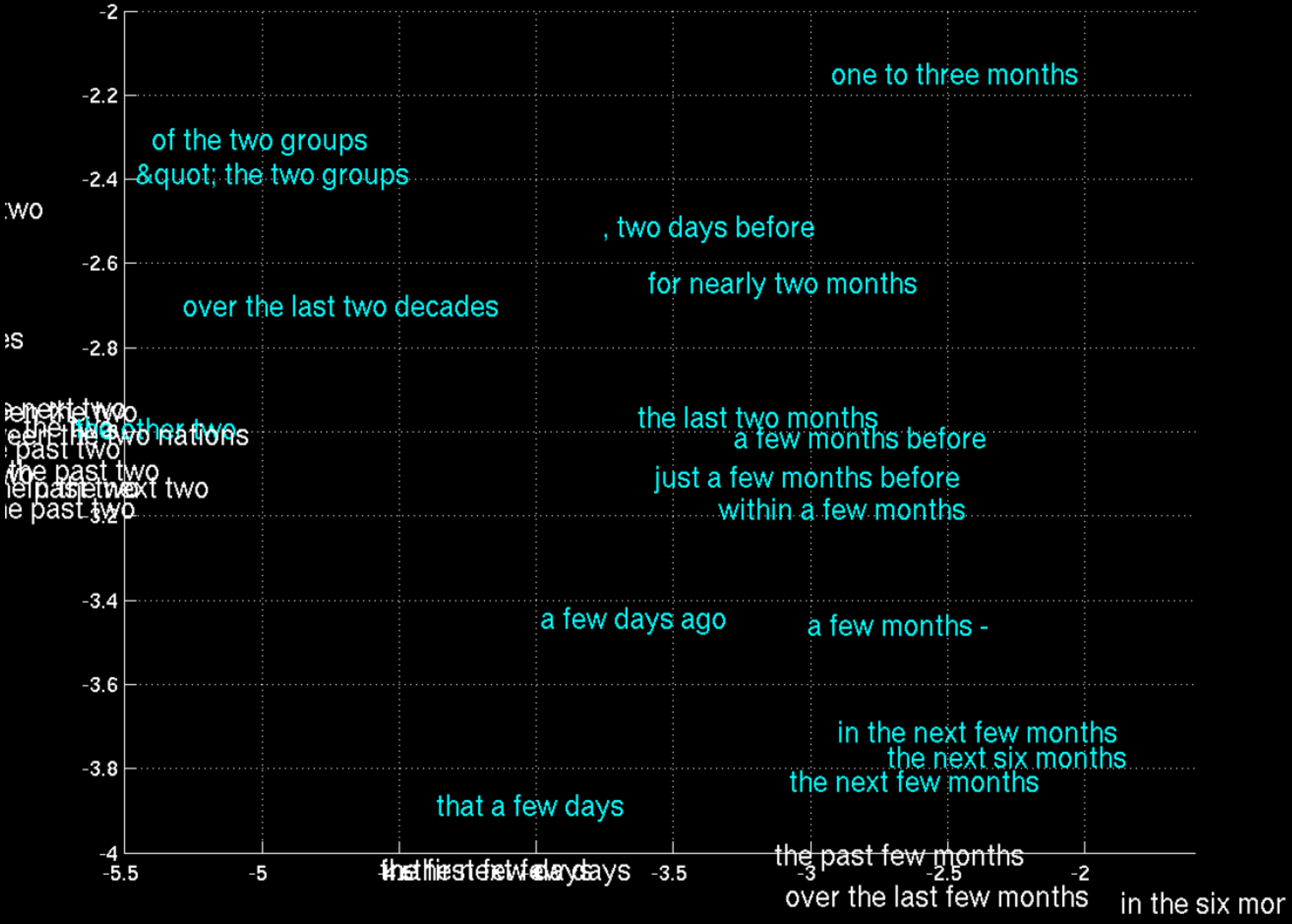
Zaremba & Sutskever (2015) Learning to Execute



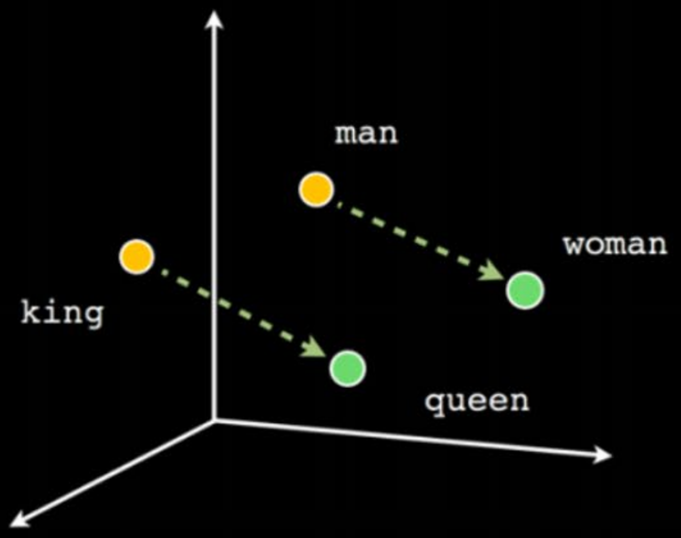
Cho et al. (2014) Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation



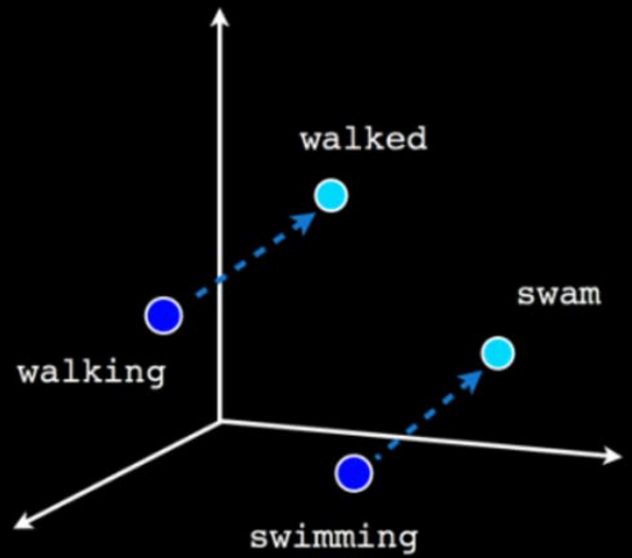
Cho et al. (2014) Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation



Cho et al. (2014) Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation

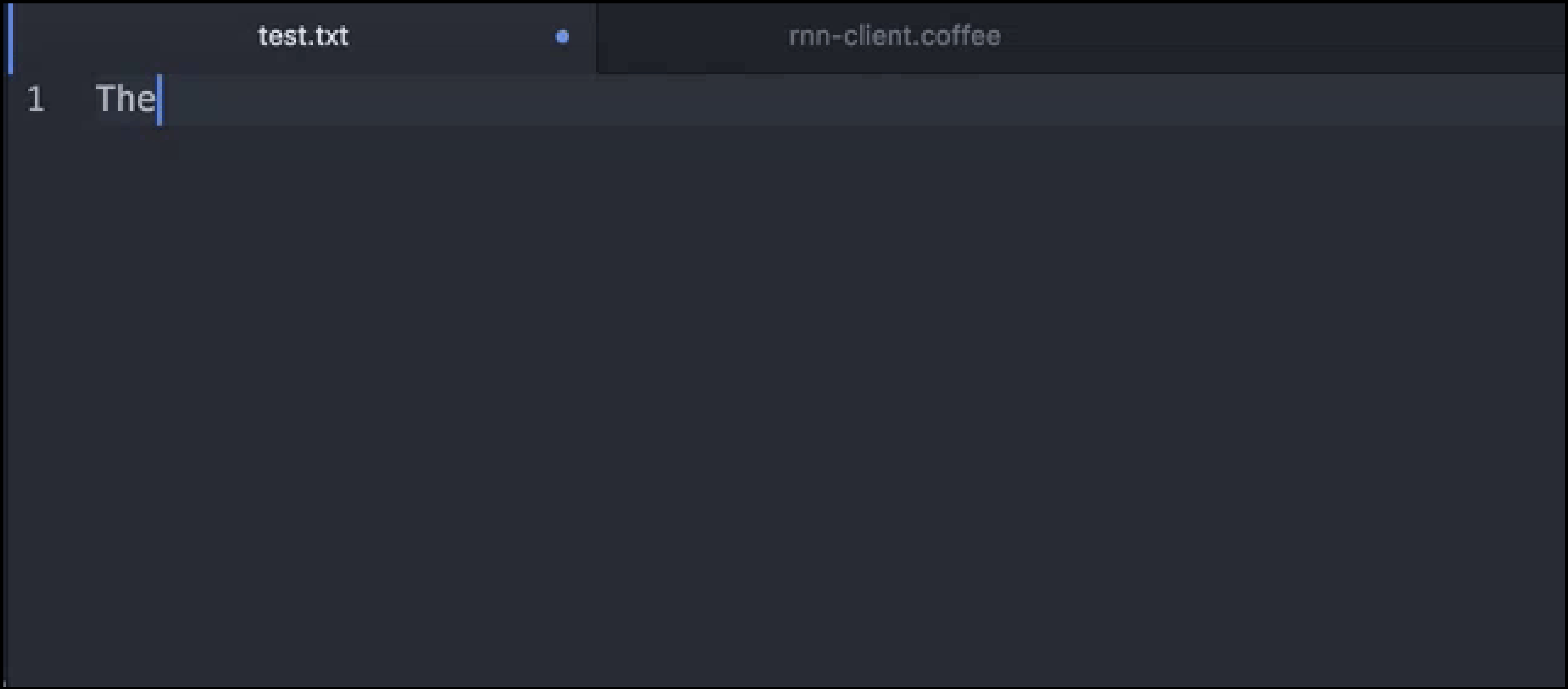


Male-Female



Verb tense

Country-Capital



Sloan (2016) rnn-writer – github.com/robinsloan/rnn-writer

SYSTEM PROMPT (HUMAN-WRITTEN)

In a shocking finding, scientist discovered a herd of unicorns living in a remote, previously unexplored valley, in the Andes Mountains. Even more surprising to the researchers was the fact that the unicorns spoke perfect English.

MODEL COMPLETION (MACHINE-WRITTEN, 10 TRIES)

The scientist named the population, after their distinctive horn, Ovid's Unicorn. These four-horned, silver-white unicorns were previously unknown to science.

Now, after almost two centuries, the mystery of what sparked this odd phenomenon is finally solved.

Dr. Jorge Pérez, an evolutionary biologist from the University of La Paz, and several companions, were exploring the Andes Mountains when they found a small valley, with no other animals or humans. Pérez noticed that the valley had what appeared to be a natural fountain, surrounded by two peaks of rock and silver snow.

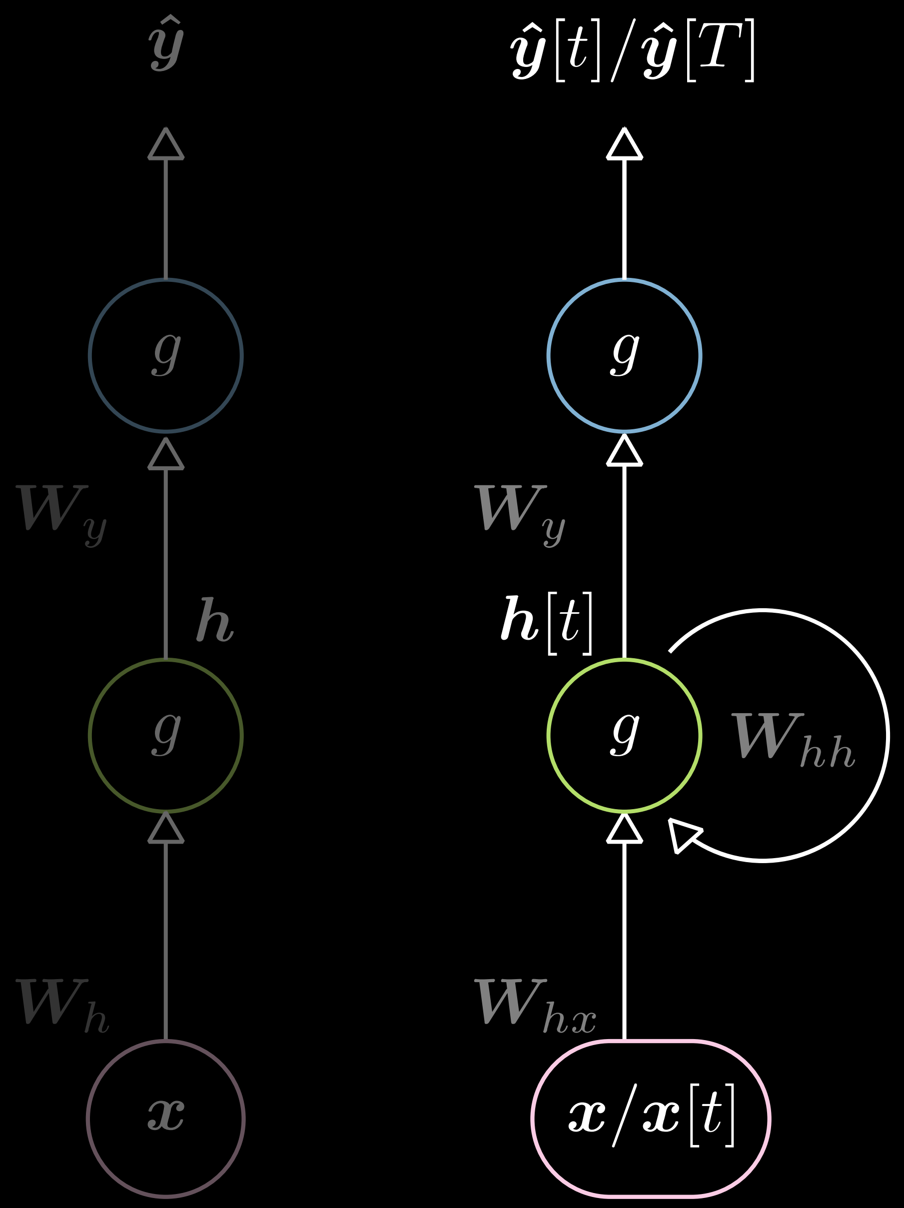
Pérez and the others then ventured further into the valley. "By the time we reached the top of one peak, the water looked blue, with some crystals on top," said Pérez.

Pérez and his friends were astonished to see the unicorn herd. These creatures could be seen from the air without having to move too much to see them – they were so close they could touch their horns.

While examining these bizarre creatures the scientists discovered that the creatures also spoke some fairly regular English. Pérez stated, "We can see, for example, that they have a common 'language,' something like a dialect or dialectic."

RNN training

Back propagation through time (BPTT)



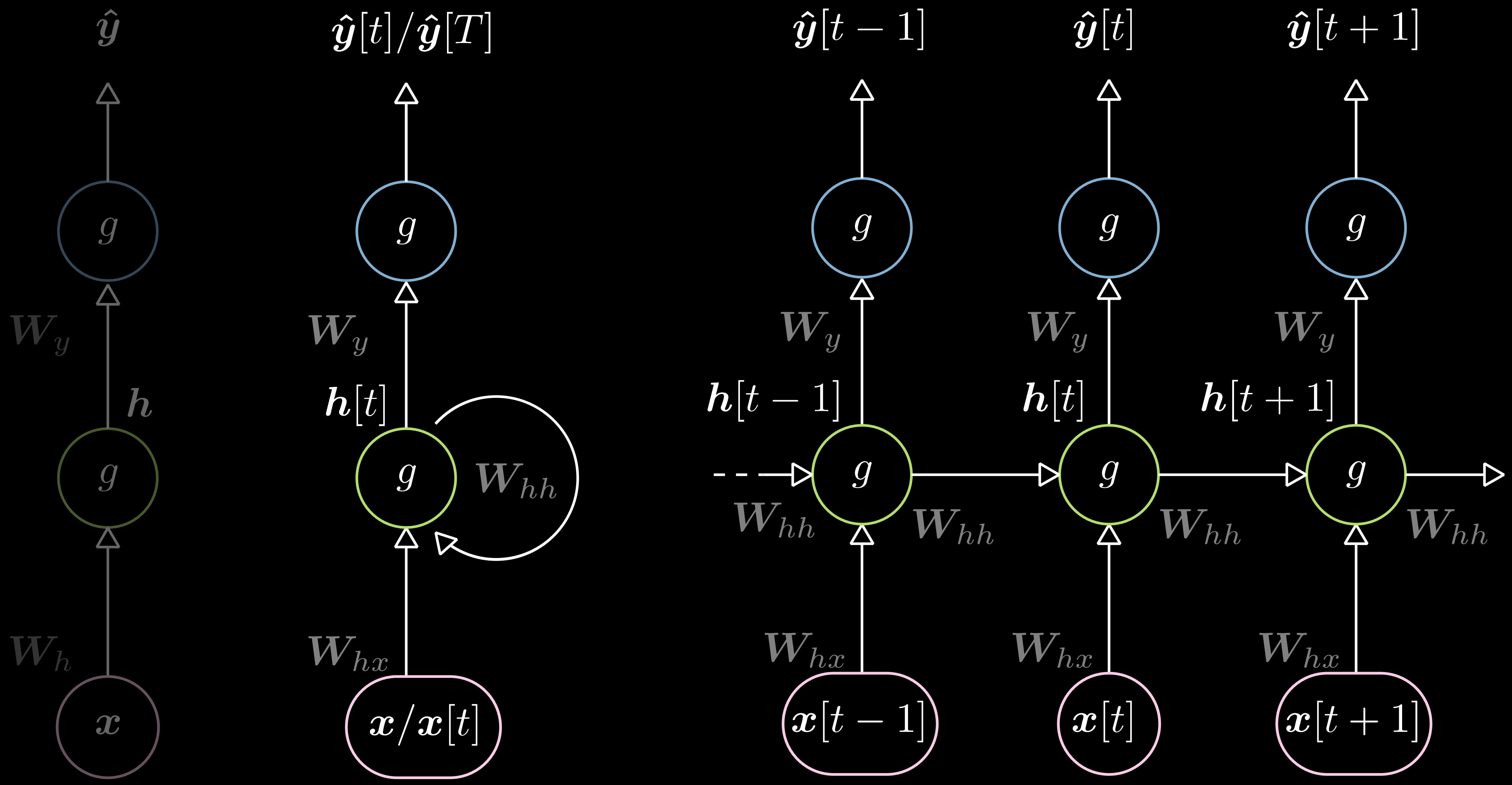
$$\boldsymbol{h} = f(\boldsymbol{W_h x} + \boldsymbol{b_h})$$

$$\hat{\boldsymbol{y}} = g(\boldsymbol{W_y h} + \boldsymbol{b_y})$$

$$\boldsymbol{h}[t] = g\left(\boldsymbol{W_h}\begin{bmatrix}\boldsymbol{x}[t] \\ \boldsymbol{h}[t-1]\end{bmatrix} + \boldsymbol{b_h}\right)$$

$$\boldsymbol{h}[0] \doteq \boldsymbol{0}, \boldsymbol{W_h} \doteq \begin{bmatrix}\boldsymbol{W_{hx}} & \boldsymbol{W_{hh}}\end{bmatrix}$$

$$\hat{\boldsymbol{y}}[t] = g(\boldsymbol{W_y h}[t] + \boldsymbol{b_y})$$



$\hat{\boldsymbol{y}}$
$\boldsymbol{W}_y$
$\boldsymbol{h}$
$\boldsymbol{W}_h$
$\boldsymbol{x}$
g
g
$\hat{\boldsymbol{y}}[t]/\hat{\boldsymbol{y}}[T]$
$\boldsymbol{W}_y$
$\boldsymbol{h}[t]$
$\boldsymbol{W}_{hh}$
$\boldsymbol{W}_{hx}$
$\boldsymbol{x}/\boldsymbol{x}[t]$
g
g
$\hat{\boldsymbol{y}}[t-1]$
$\hat{\boldsymbol{y}}[t]$
$\hat{\boldsymbol{y}}[t+1]$
$\boldsymbol{W}_y$
$\boldsymbol{W}_y$
$\boldsymbol{W}_y$
$\boldsymbol{h}[t-1]$
$\boldsymbol{h}[t]$
$\boldsymbol{h}[t+1]$
$\boldsymbol{W}_{hh}$
$\boldsymbol{W}_{hh}$
$\boldsymbol{W}_{hh}$
$\boldsymbol{W}_{hh}$
$\boldsymbol{W}_{hx}$
$\boldsymbol{W}_{hx}$
$\boldsymbol{W}_{hx}$
$\boldsymbol{x}[t-1]$
$\boldsymbol{x}[t]$
$\boldsymbol{x}[t+1]$

Training example

Language modelling

Batch-ification

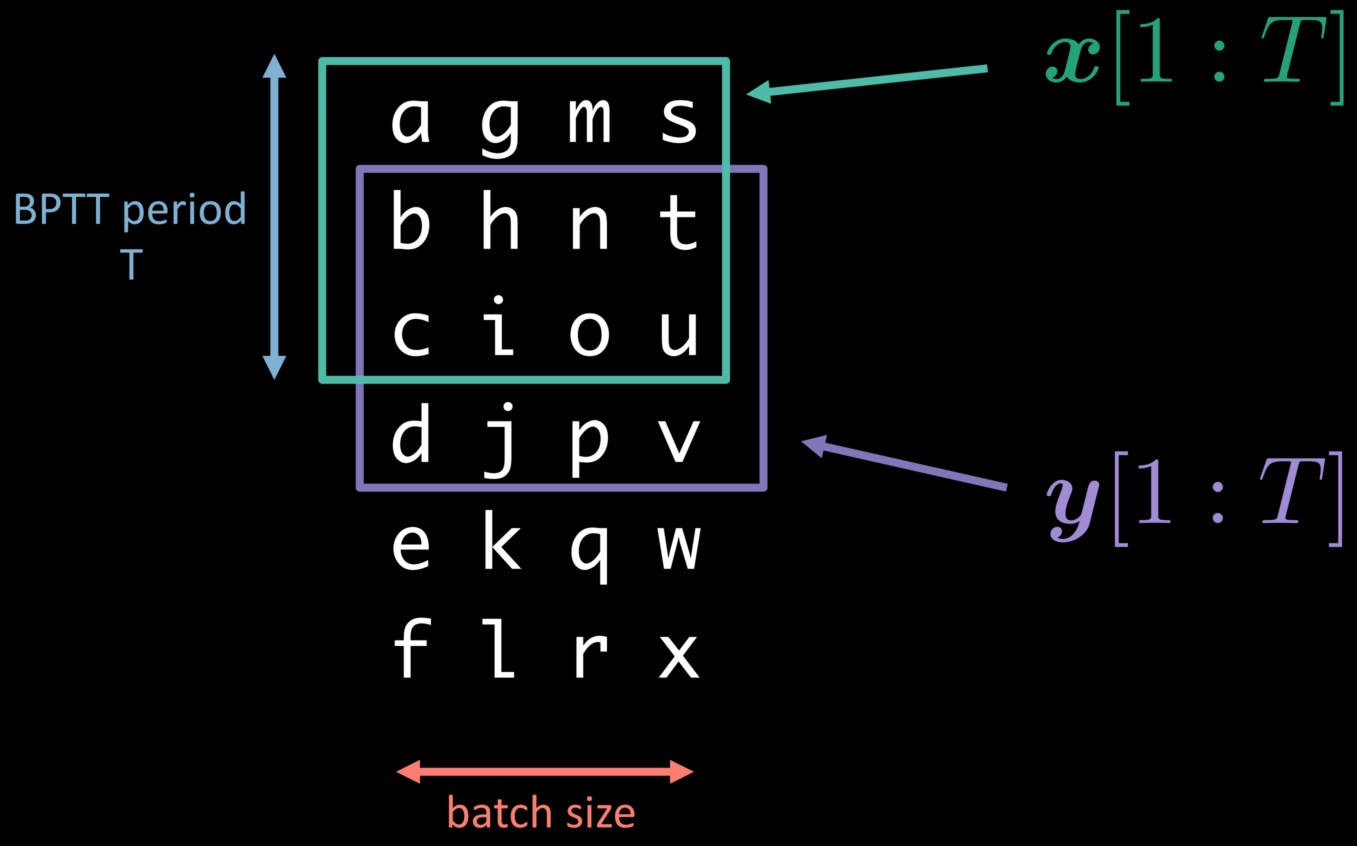
abcdefghijklmnopqrstuvwxyz

a	g	m	s
b	h	n	t
c	i	o	u
d	j	p	v
e	k	q	w
f	l	r	x

batch size

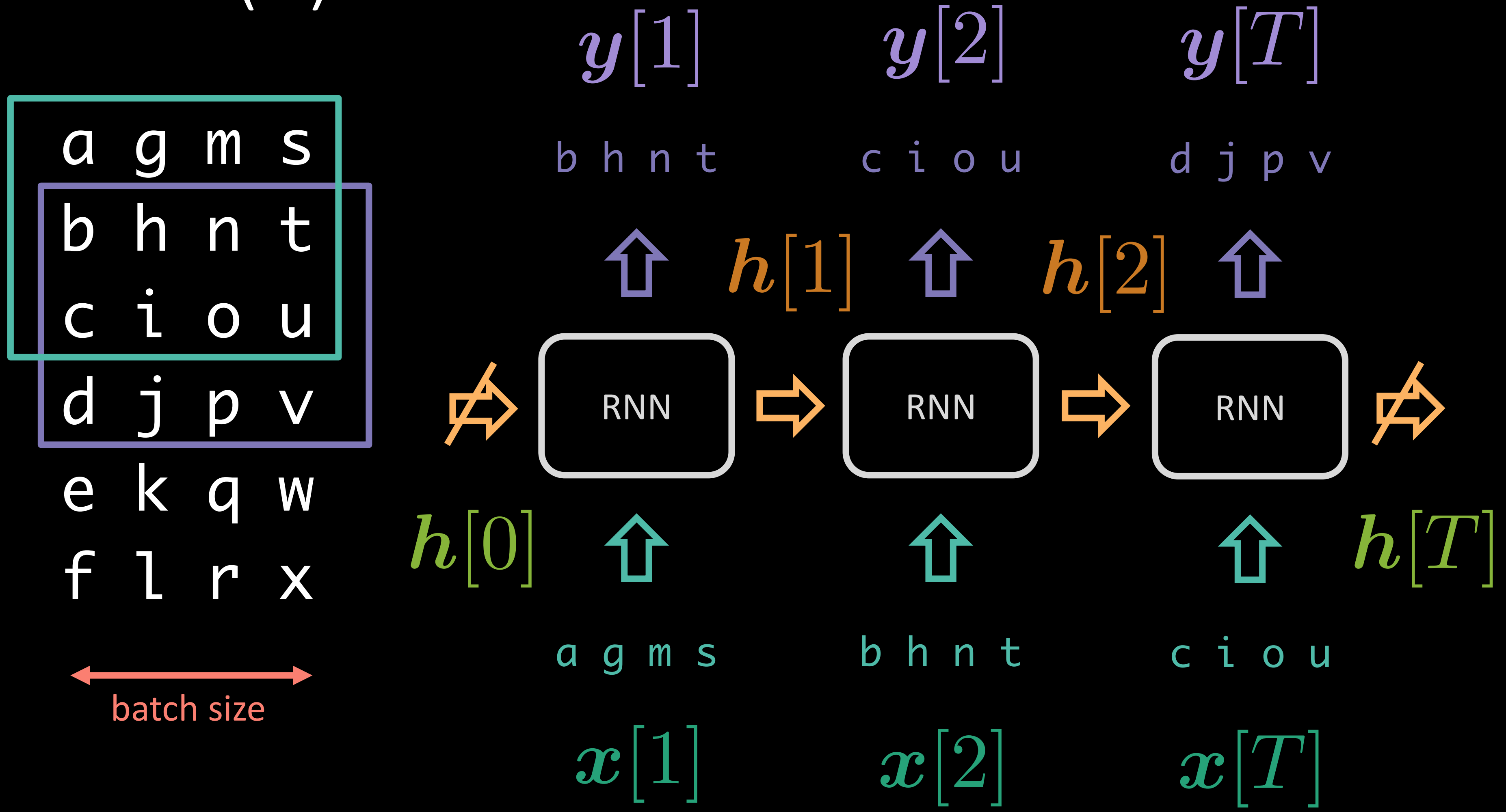
Check `word_language_model @ github.com/pytorch/examples/`

Get batch (I)



Check `word_language_model @ github.com/pytorch/examples/`

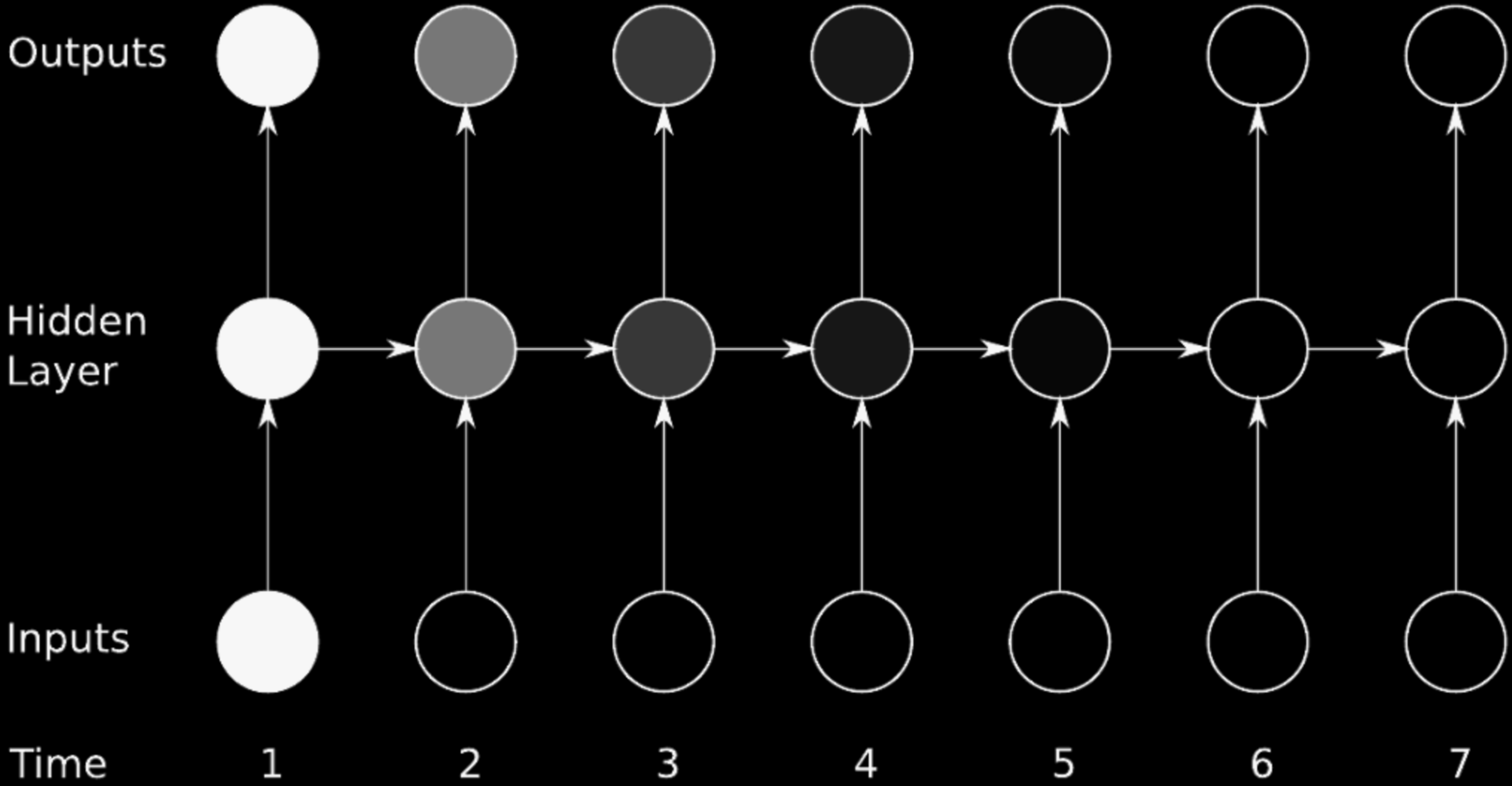
Get batch (II)



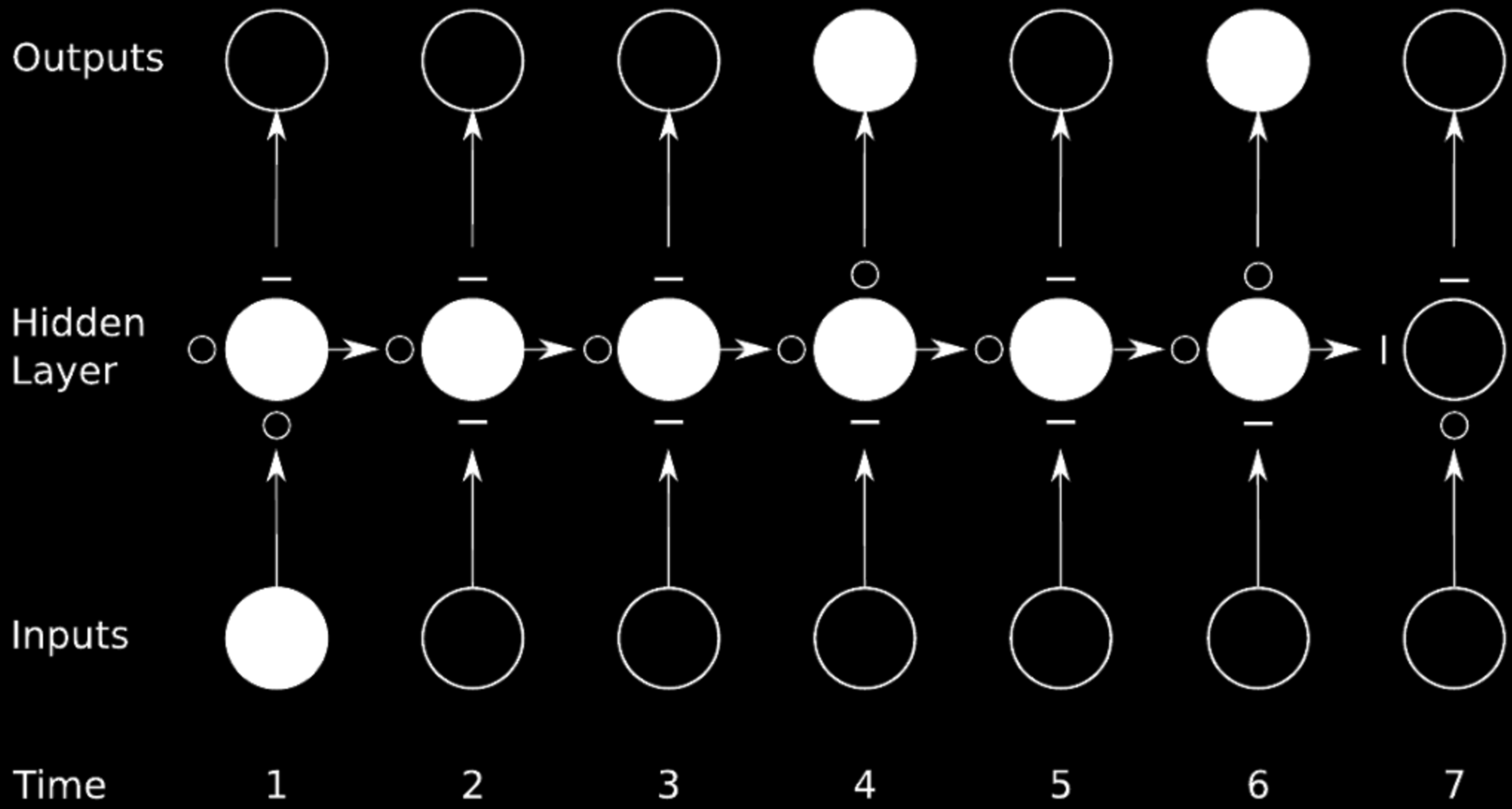
Check `word_language_model` @ `github.com/pytorch/examples/`

Vanishing & exploding gradients

Limitations of temporally deep nets



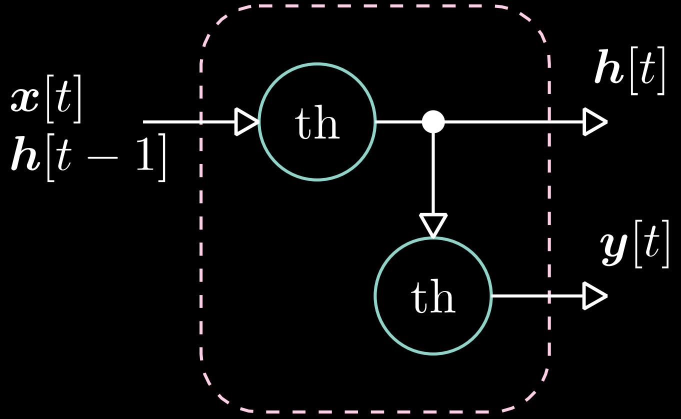
Graves (2012) Supervised sequence labelling



Graves (2012) Supervised sequence labelling

Long Short-Term Memory

Gated RNN



$$\boldsymbol{h}[t] = g\left(\boldsymbol{W}_h \begin{bmatrix} \boldsymbol{x}[t] \\ \boldsymbol{h}[t-1] \end{bmatrix} + \boldsymbol{b}_h\right)$$

$$\hat{\boldsymbol{y}}[t] = g(\boldsymbol{W}_y \boldsymbol{h}[t] + \boldsymbol{b}_y)$$

$$\boldsymbol{i}[t] = \sigma\left(\boldsymbol{W}_i \begin{bmatrix} \boldsymbol{x}[t] \\ \boldsymbol{h}[t-1] \end{bmatrix} + \boldsymbol{b}_i\right)$$

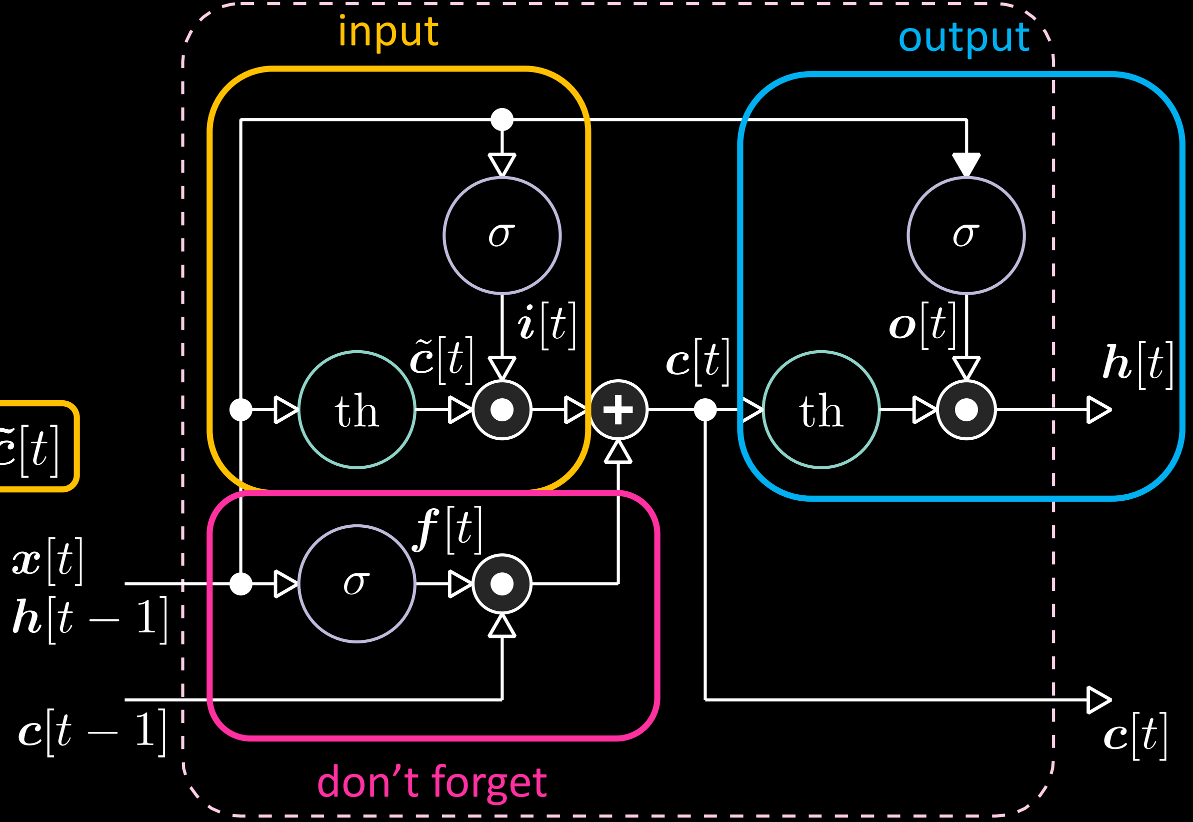
$$\boldsymbol{f}[t] = \sigma\left(\boldsymbol{W}_f \begin{bmatrix} \boldsymbol{x}[t] \\ \boldsymbol{h}[t-1] \end{bmatrix} + \boldsymbol{b}_f\right)$$

$$\boldsymbol{o}[t] = \sigma\left(\boldsymbol{W}_o \begin{bmatrix} \boldsymbol{x}[t] \\ \boldsymbol{h}[t-1] \end{bmatrix} + \boldsymbol{b}_o\right)$$

$$\tilde{\boldsymbol{c}}[t] = \tanh\left(\boldsymbol{W}_c \begin{bmatrix} \boldsymbol{x}[t] \\ \boldsymbol{h}[t-1] \end{bmatrix} + \boldsymbol{b}_c\right)$$

$$\boldsymbol{c}[t] = \boxed{\boldsymbol{f}[t] \odot \boldsymbol{c}[t-1]} + \boxed{\boldsymbol{i}[t] \odot \tilde{\boldsymbol{c}}[t]}$$

$$\boldsymbol{h}[t] = \boxed{\boldsymbol{o}[t]} \odot \boxed{\tanh(\boldsymbol{c}[t])}$$



Controlling the $\boxed{\text{output}}$ - OFF

Saturated sigmoid: green circle = 1, red circle = 0

$$\boldsymbol{i}[t] = \sigma\left(\boldsymbol{W}_i \begin{bmatrix} \boldsymbol{x}[t] \\ \boldsymbol{h}[t-1] \end{bmatrix} + \boldsymbol{b}_i\right)$$

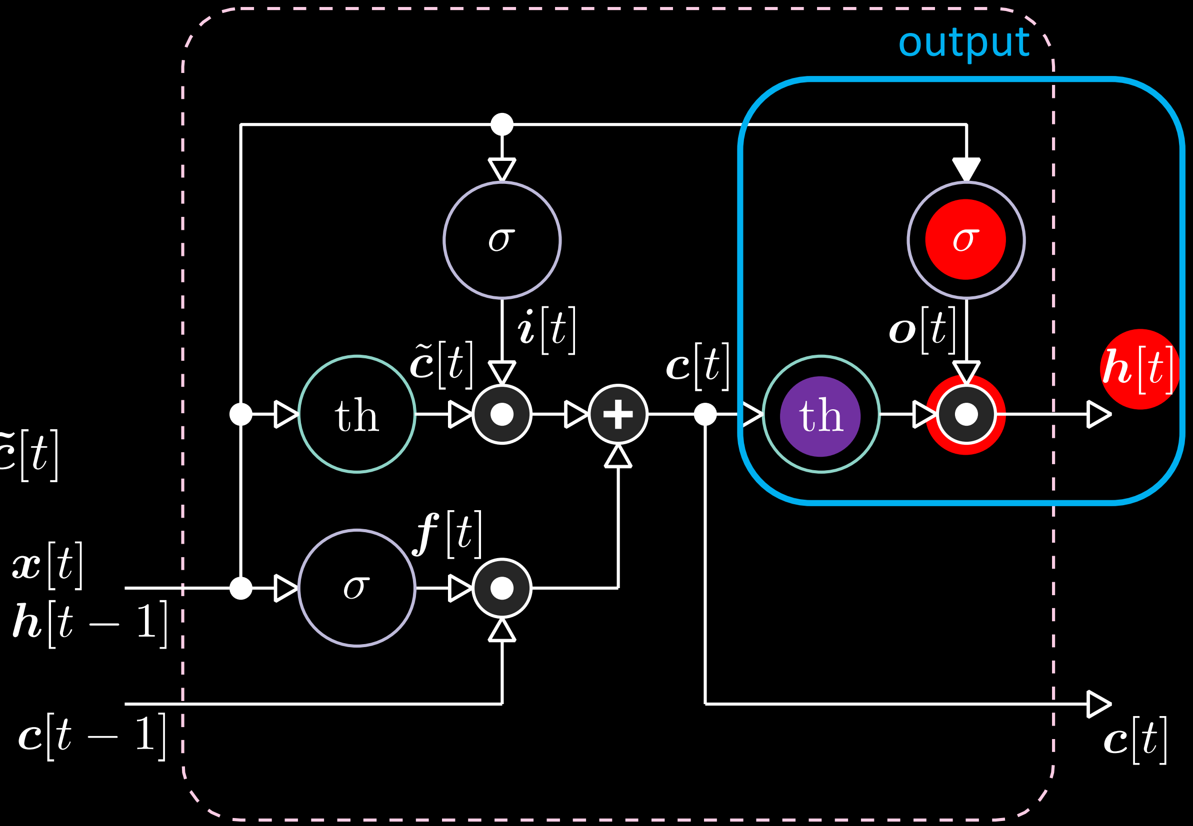
$$\boldsymbol{f}[t] = \sigma\left(\boldsymbol{W}_f \begin{bmatrix} \boldsymbol{x}[t] \\ \boldsymbol{h}[t-1] \end{bmatrix} + \boldsymbol{b}_f\right)$$

$$\boxed{\boldsymbol{o}[t] = \sigma\left(\boldsymbol{W}_o \begin{bmatrix} \boldsymbol{x}[t] \\ \boldsymbol{h}[t-1] \end{bmatrix} + \boldsymbol{b}_o\right)}$$

$$\tilde{\boldsymbol{c}}[t] = \tanh\left(\boldsymbol{W}_c \begin{bmatrix} \boldsymbol{x}[t] \\ \boldsymbol{h}[t-1] \end{bmatrix} + \boldsymbol{b}_c\right)$$

$$\boldsymbol{c}[t] = \boldsymbol{f}[t] \odot \boldsymbol{c}[t-1] + \boldsymbol{i}[t] \odot \tilde{\boldsymbol{c}}[t]$$

$$\boldsymbol{h}[t] = \boxed{\boldsymbol{o}[t] \odot \tanh(\boldsymbol{c}[t])}$$



Controlling the $\boxed{\text{output}}$ - ON

Saturated sigmoid $\begin{cases} \text{green} = 1 \\ \text{red} = 0 \end{cases}$

$$\boldsymbol{i}[t] = \sigma\left(\boldsymbol{W}_i \begin{bmatrix}\boldsymbol{x}[t] \\ \boldsymbol{h}[t-1]\end{bmatrix} + \boldsymbol{b}_i\right)$$

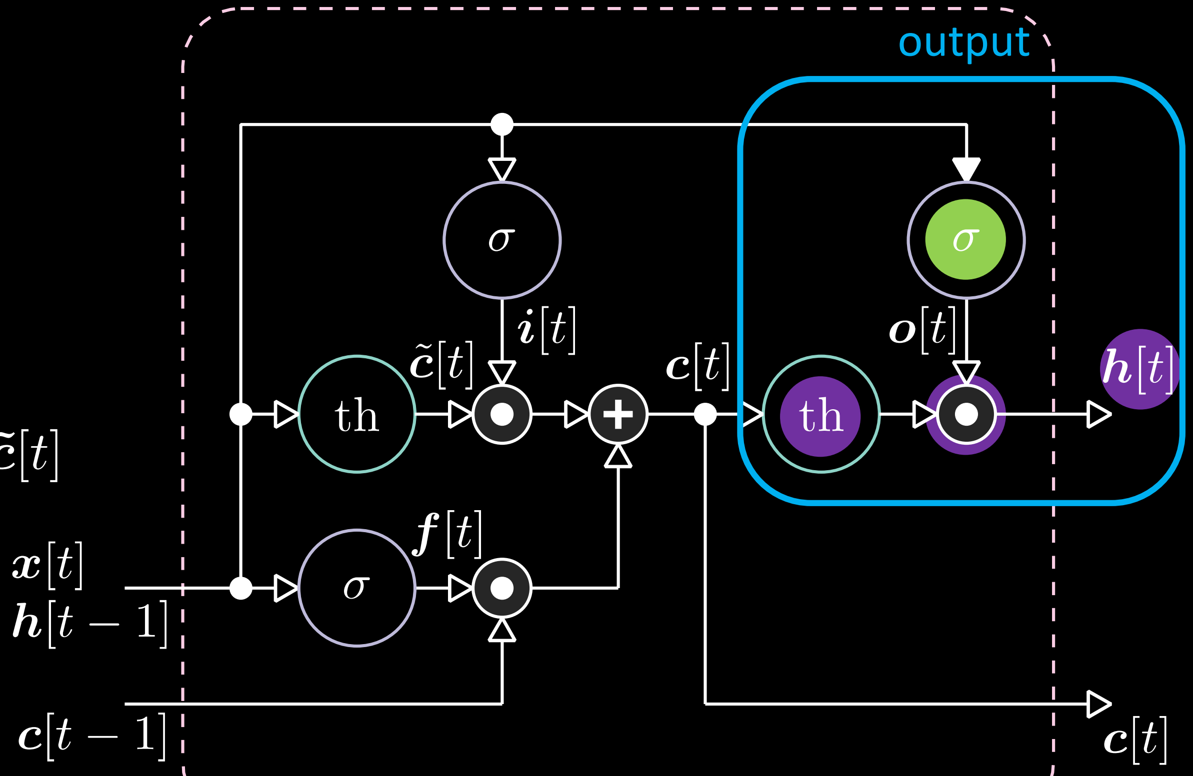
$$\boldsymbol{f}[t] = \sigma\left(\boldsymbol{W}_f \begin{bmatrix}\boldsymbol{x}[t] \\ \boldsymbol{h}[t-1]\end{bmatrix} + \boldsymbol{b}_f\right)$$

$$\boxed{\boldsymbol{o}[t] = \sigma\left(\boldsymbol{W}_o \begin{bmatrix}\boldsymbol{x}[t] \\ \boldsymbol{h}[t-1]\end{bmatrix} + \boldsymbol{b}_o\right)}$$

$$\tilde{\boldsymbol{c}}[t] = \tanh\left(\boldsymbol{W}_c \begin{bmatrix}\boldsymbol{x}[t] \\ \boldsymbol{h}[t-1]\end{bmatrix} + \boldsymbol{b}_c\right)$$

$$\boldsymbol{c}[t] = \boldsymbol{f}[t] \odot \boldsymbol{c}[t-1] + \boldsymbol{i}[t] \odot \tilde{\boldsymbol{c}}[t]$$

$$\boldsymbol{h}[t] = \boxed{\boldsymbol{o}[t] \odot \tanh(\boldsymbol{c}[t])}$$



Controlling the memory - reset

Saturated sigmoid
= 1
= 0

$$\boldsymbol{i}[t] = \sigma\left(\boldsymbol{W}_i \begin{bmatrix}\boldsymbol{x}[t] \\ \boldsymbol{h}[t-1]\end{bmatrix} + \boldsymbol{b}_i\right)$$

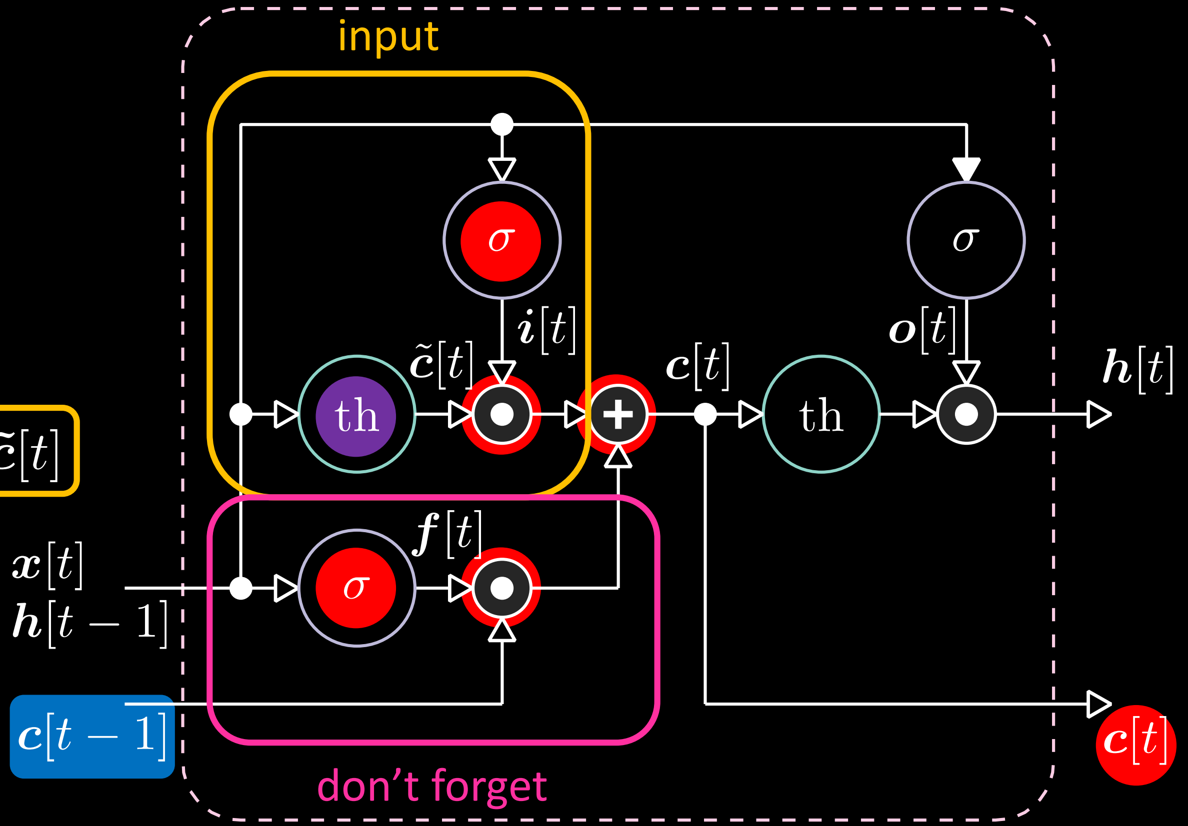
$$\boldsymbol{f}[t] = \sigma\left(\boldsymbol{W}_f \begin{bmatrix}\boldsymbol{x}[t] \\ \boldsymbol{h}[t-1]\end{bmatrix} + \boldsymbol{b}_f\right)$$

$$\boldsymbol{o}[t] = \sigma\left(\boldsymbol{W}_o \begin{bmatrix}\boldsymbol{x}[t] \\ \boldsymbol{h}[t-1]\end{bmatrix} + \boldsymbol{b}_o\right)$$

$$\tilde{\boldsymbol{c}}[t] = \tanh\left(\boldsymbol{W}_c \begin{bmatrix}\boldsymbol{x}[t] \\ \boldsymbol{h}[t-1]\end{bmatrix} + \boldsymbol{b}_c\right)$$

$$\boldsymbol{c}[t] = \boxed{\boldsymbol{f}[t] \odot \boldsymbol{c}[t-1]} + \boxed{\boldsymbol{i}[t] \odot \tilde{\boldsymbol{c}}[t]}$$

$$\boldsymbol{h}[t] = \boldsymbol{o}[t] \odot \tanh(\boldsymbol{c}[t])$$



Controlling the memory - keep

Saturated sigmoid: green $= 1$, red $= 0$

$$\boldsymbol{i}[t] = \sigma\left(\boldsymbol{W}_i \begin{bmatrix} \boldsymbol{x}[t] \\ \boldsymbol{h}[t-1] \end{bmatrix} + \boldsymbol{b}_i\right)$$

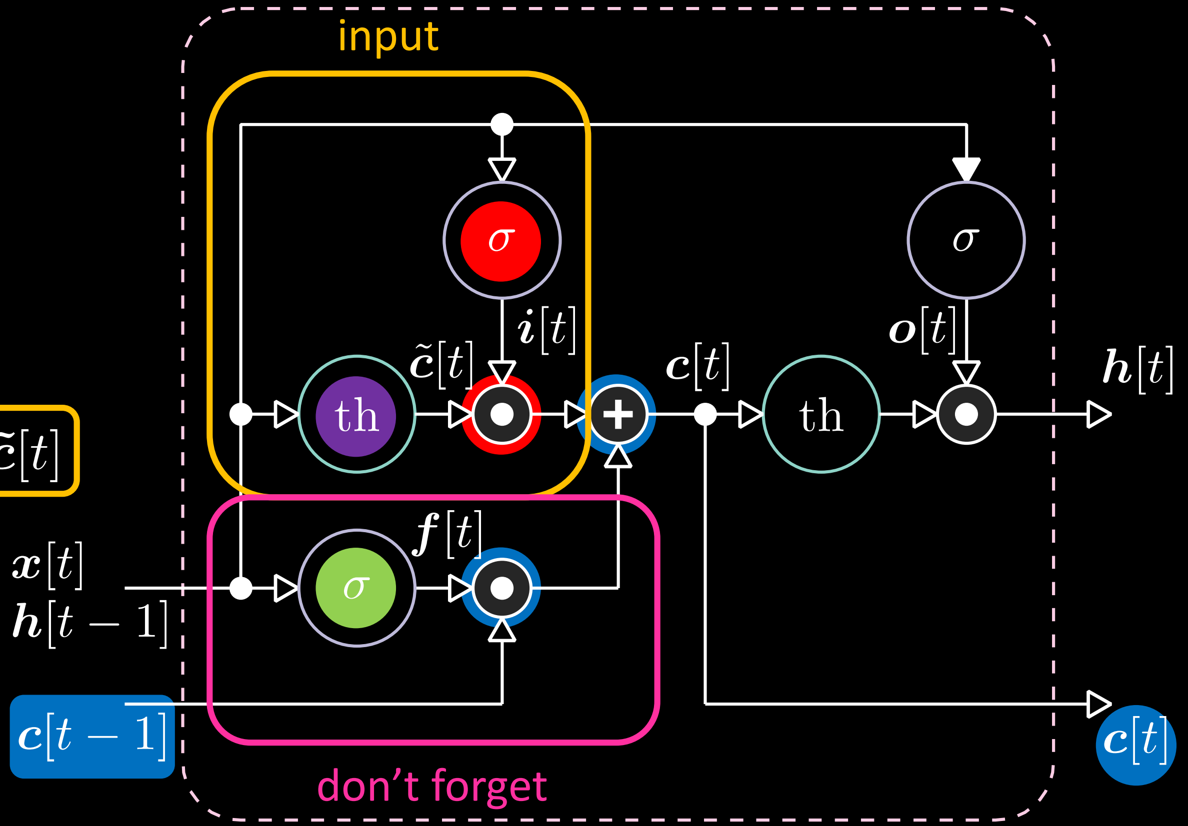
$$\boldsymbol{f}[t] = \sigma\left(\boldsymbol{W}_f \begin{bmatrix} \boldsymbol{x}[t] \\ \boldsymbol{h}[t-1] \end{bmatrix} + \boldsymbol{b}_f\right)$$

$$\boldsymbol{o}[t] = \sigma\left(\boldsymbol{W}_o \begin{bmatrix} \boldsymbol{x}[t] \\ \boldsymbol{h}[t-1] \end{bmatrix} + \boldsymbol{b}_o\right)$$

$$\tilde{\boldsymbol{c}}[t] = \tanh\left(\boldsymbol{W}_c \begin{bmatrix} \boldsymbol{x}[t] \\ \boldsymbol{h}[t-1] \end{bmatrix} + \boldsymbol{b}_c\right)$$

$$\boldsymbol{c}[t] = \boxed{\boldsymbol{f}[t] \odot \boldsymbol{c}[t-1]} + \boxed{\boldsymbol{i}[t] \odot \tilde{\boldsymbol{c}}[t]}$$

$$\boldsymbol{h}[t] = \boldsymbol{o}[t] \odot \tanh(\boldsymbol{c}[t])$$



Controlling the $\boxed{\text{memory}}$ - write

Saturated sigmoid
- (green) = 1
- (red) = 0

$$\boxed{\boldsymbol{i}[t] = \sigma\left(\boldsymbol{W}_i \begin{bmatrix} \boldsymbol{x}[t] \\ \boldsymbol{h}[t-1] \end{bmatrix} + \boldsymbol{b}_i\right)}$$

$$\boxed{\boldsymbol{f}[t] = \sigma\left(\boldsymbol{W}_f \begin{bmatrix} \boldsymbol{x}[t] \\ \boldsymbol{h}[t-1] \end{bmatrix} + \boldsymbol{b}_f\right)}$$

$$\boldsymbol{o}[t] = \sigma\left(\boldsymbol{W}_o \begin{bmatrix} \boldsymbol{x}[t] \\ \boldsymbol{h}[t-1] \end{bmatrix} + \boldsymbol{b}_o\right)$$

$$\tilde{\boldsymbol{c}}[t] = \tanh\left(\boldsymbol{W}_c \begin{bmatrix} \boldsymbol{x}[t] \\ \boldsymbol{h}[t-1] \end{bmatrix} + \boldsymbol{b}_c\right)$$

$$\boldsymbol{c}[t] = \boxed{\boldsymbol{f}[t] \odot \boldsymbol{c}[t-1]} + \boxed{\boldsymbol{i}[t] \odot \tilde{\boldsymbol{c}}[t]}$$

$$\boldsymbol{h}[t] = \boldsymbol{o}[t] \odot \tanh(\boldsymbol{c}[t])$$