

Recurrent Neural Network

MLRS 2019

Rachel Hu

Amazon AI

Outline

- Sequence models and language models
- Recurrent neural networks (RNN)
- GRU and LSTM
- Deep RNNs, Bi-RNNs

Last but not the least ...

AWS Setting-up

1. The region

Asia Pacific (Mumbai)

Asia Pacific (Osaka-Local)

Asia Pacific (Seoul)

Asia Pacific (Singapore)

Asia Pacific (Sydney)

Asia Pacific (Tokyo)

AWS Setting-up - Billing

2. Billing: Don't forget to stop it. (right click your instance to stop)

The screenshot shows the AWS EC2 Dashboard. On the left, there's a sidebar with options like EC2 Dashboard, Events, Tags, Reports, Limits, INSTANCES, Instances, Launch Templates, Spot Requests, Reserved Instances, Dedicated Hosts, Scheduled Instances, and Capacity Reservations. The Instances option is currently selected. In the main area, there's a search bar and a table with columns for Name, Instance ID, Instance Type, and Availability Zone. A row for an instance named 'rachel_mlrs' is selected. A context menu is open over this instance, listing options: Connect, Get Windows Password, Create Template From Instance, Launch More Like This, Instance State (with sub-options Start, Stop, Stop - Hibernate, Reboot, Terminate), Instance Settings, Image, Networking, and CloudWatch Monitoring. The 'Stop' option under Instance State is highlighted.

AWS Setting-up - Billing

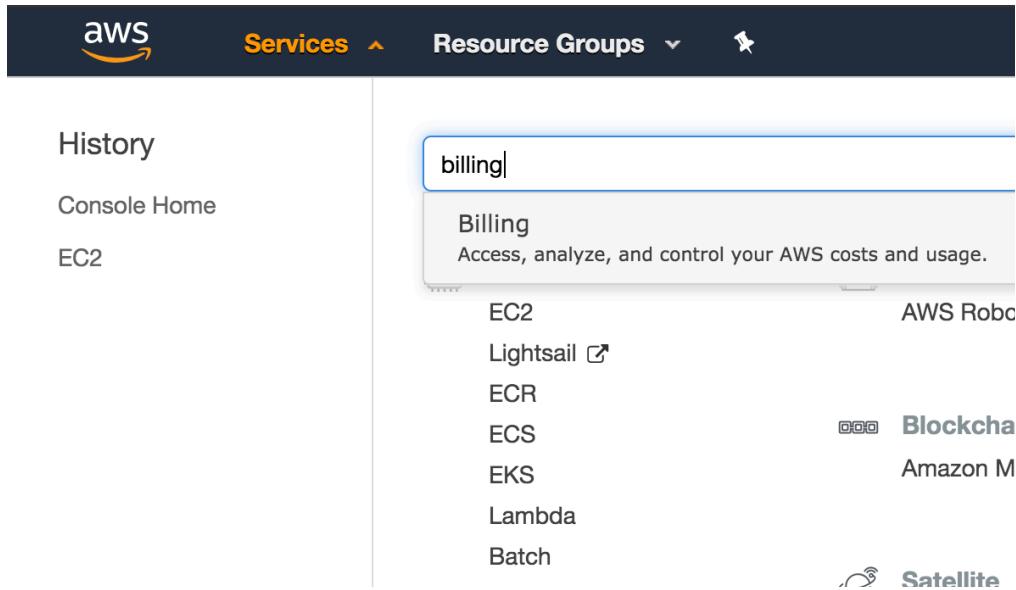
2. Billing

“Stopping” an instance means that you will be able to start it again. (However, stopped instances will still be billed a small amount for the hard disk space retained.)

“Terminate” deletes all data associated with it.

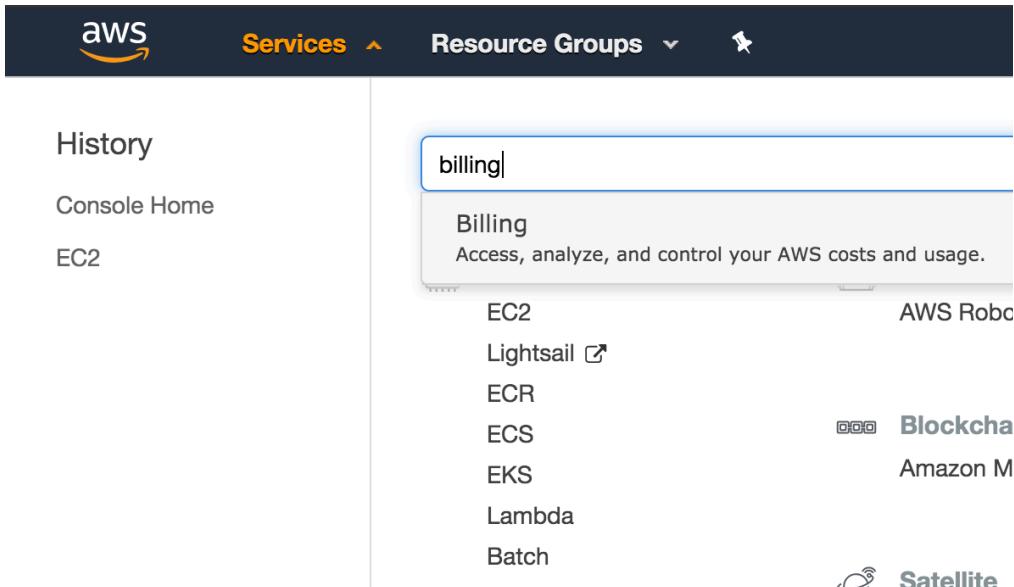
AWS Setting-up - Billing

Enter “billing” in the search box



AWS Setting-up - Billing

Enter “billing” in the search box



Home

Cost Management

Cost Explorer

Budgets

Budgets Reports

Cost & Usage Reports

Cost allocation tags

Billing

Bills

Payment history

Credits

Preferences

Billing preferences

Payment methods

Consolidated billing

Tax settings

Billing & Cost Management Dashboard



Getting Started with AWS Billing & Cost Management

- Manage your costs and usage using [AWS Budgets](#)
- Visualize your cost drivers and usage trends via [Cost Explorer](#)
- Dive deeper into your costs using the [Cost and Usage Reports](#) with Athena integration
- Learn more:** Check out the [AWS What's New webpage](#)

Do you have Reserved Instances (RIs)?

- Access the RI Utilization & Coverage reports—and RI purchase recommendations—via [Cost Explorer](#).

Spend Summary

Cost Explorer

Welcome to the AWS Account Billing console. Your last month and month-to-date costs appear below.

Current month-to-date balance for August 2019

\$0.00



AWS Setting-up - Billing

If you forget to turn off the instance...

AWS DeepRacer



Who is this???

<https://aws.amazon.com/deepracer/>

AWS DeepRacer League



You can access the [AWS DeepRacer 3D racing simulator](#) in the AWS console to train your models, evaluate them, and take part in the AWS DeepRacer League.

<https://aws.amazon.com/deepracer/>

AWS DeepRacer League



<https://aws.amazon.com/deepracer/>

numpy.d2l.ai

How do I get started with AWS DeepRacer simulator?

The [AWS DeepRacer simulator](#) provides a tutorial to get you started.

The AWS DeepRacer Developer Documentation provides additional details on [building your first model](#) and also [how to improve your models](#).



Recommended Courses/Textbooks/Resources (Online)

<https://github.com/goldmermaid/mlrs>

Title	Website	Notes
Introduction to Deep Learning	Course	Introductory to medium level deep learning course (that we taught at UC Berkeley). Lecture videos, slides, homework, etc. are all available online.
Introduction to Machine Learning	Course	Introductory level machine learning course. Lecture videos, slides, homework, etc. are all available online.
The Foundations of Data Science	Course	Introductory level machine learning course. Lecture videos, slides, homework, etc. are all available online.
Deep Reinforcement Learning	Course	Advanced level reinforcement learning course. Lecture videos, slides, homework, etc. are all available online.

Recommended Courses/Textbooks/Resources (Online)

<https://github.com/goldmermaid/mlrs>

Dive into Deep Learning	Textbook	Textbook with interactive jupyter notebook to get your hand dirty on deep learning.
GluonCV	GitHub	A python toolkit which provides implementations of the state-of-the-art (SOTA) deep learning models in computer vision.
GluonNLP	GitHub	A python toolkit that enables easy text preprocessing, datasets loading and neural models building to help you speed up your Natural Language Processing (NLP) research.
GluonTS	GitHub	A python toolkit for probabilistic time series modeling, which provides utilities for loading and iterating over time series datasets, state of the art models ready to be trained, and building blocks to define your own models and quickly experiment with different solutions.
MXNet Online Discussion	Website	An online protal to discuss deep learning, machine learning, how to do things efficiently in MxNet/Gluon, ask for help, or suggest new things.

Just ask online!



<https://discuss.mxnet.io/c/d2l-book>



Do you want live notifications when people reply to your posts? [Enable Notifications.](#)

D2L Book ▶ all tags ▶ **Latest** Top

Edit + New Topic

Topic

Replies Views

↳ Dive into Deep Learning - The Book

Welcome to discussions related to Dive into Deep Learning. Each section of the book has its matching discussion thread here. This should help you resolve any questions, give feedback to the authors, and discuss any relat... [read more](#)



2

302

Naive Bayes Classification



13

404

Anchor Boxes



2

150

Installation 28



36

1.7k

Linear algebra



1

168

Softmax Regression in Gluon

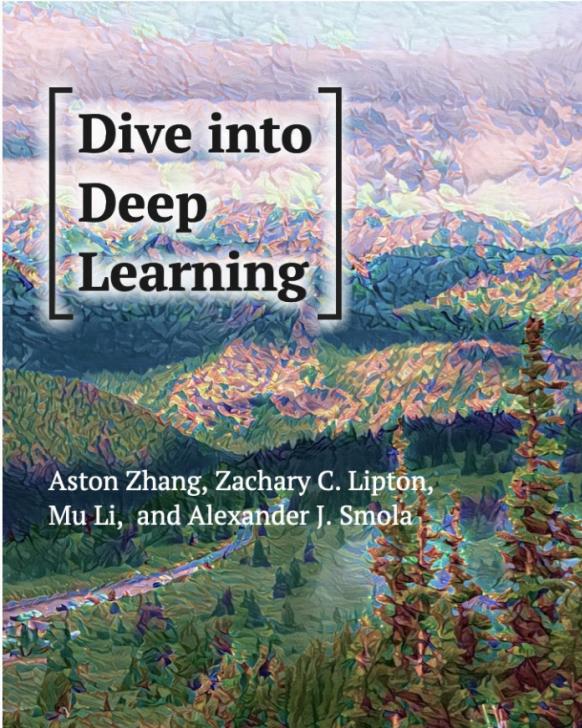


9

223

D2L Textbooks

<http://numpy.d2l.ai/>



Dive into Deep Learning

An interactive deep learning book with code, math, and discussions

Based on the NumPy interface

The contents are under revision

D2L Textbooks

<http://numpy.d2l.ai/>



[Aston Zhang](#)

Amazon Applied Scientist
UIUC Ph.D.



[Zack C. Lipton](#)

Amazon Applied Scientist
CMU Assistant Professor
UCSD Ph.D.



[Mu Li](#)

Amazon Principal Scientist
CMU Ph.D.



[Alex J. Smola](#)

Amazon VP/Distinguished
Scientist
TU Berlin Ph.D.

We thank all the [community contributors](#)
for making this open source book better for everyone.

[Contribute to the book.](#)

Last but not the least

Please reach me at rlhu@amazon.com for further cooperations, if you would like to:

- 1.teach Deep Learning with our textbook;
- 2.translate to different languages;
- 3.complain about the errors, typos, uncleanness, etc. about the book;

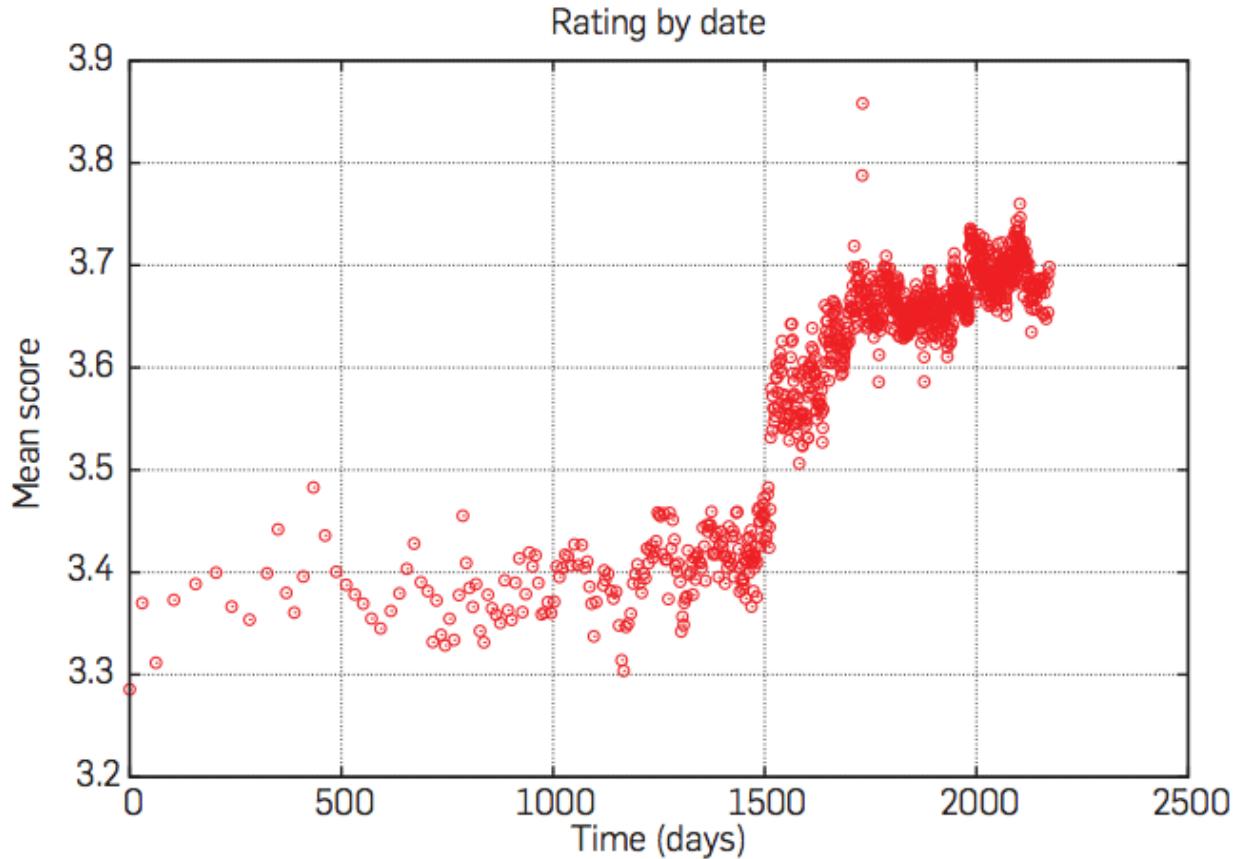
Dependent Random Variables



Data

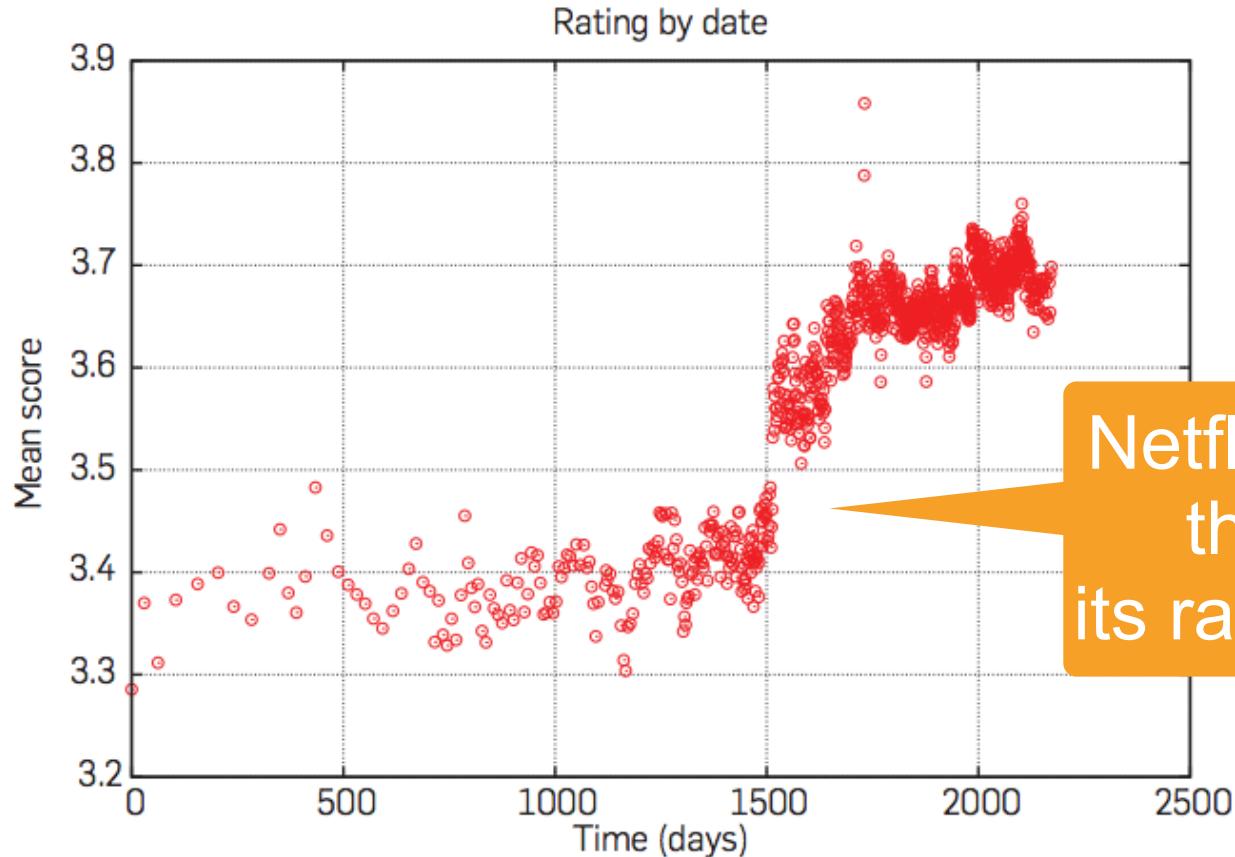
- So far ...
 - Collect observation pairs $(x_i, y_i) \sim p(x, y)$ for training
 - Estimate $y|x \sim p(y|x)$ for unseen $x' \sim p(x)$
- Examples
 - Images classification & objects recognition
 - Disease prediction
 - Housing price prediction
- **The order of the data does not matter**

Time matters (Koren, 2009)

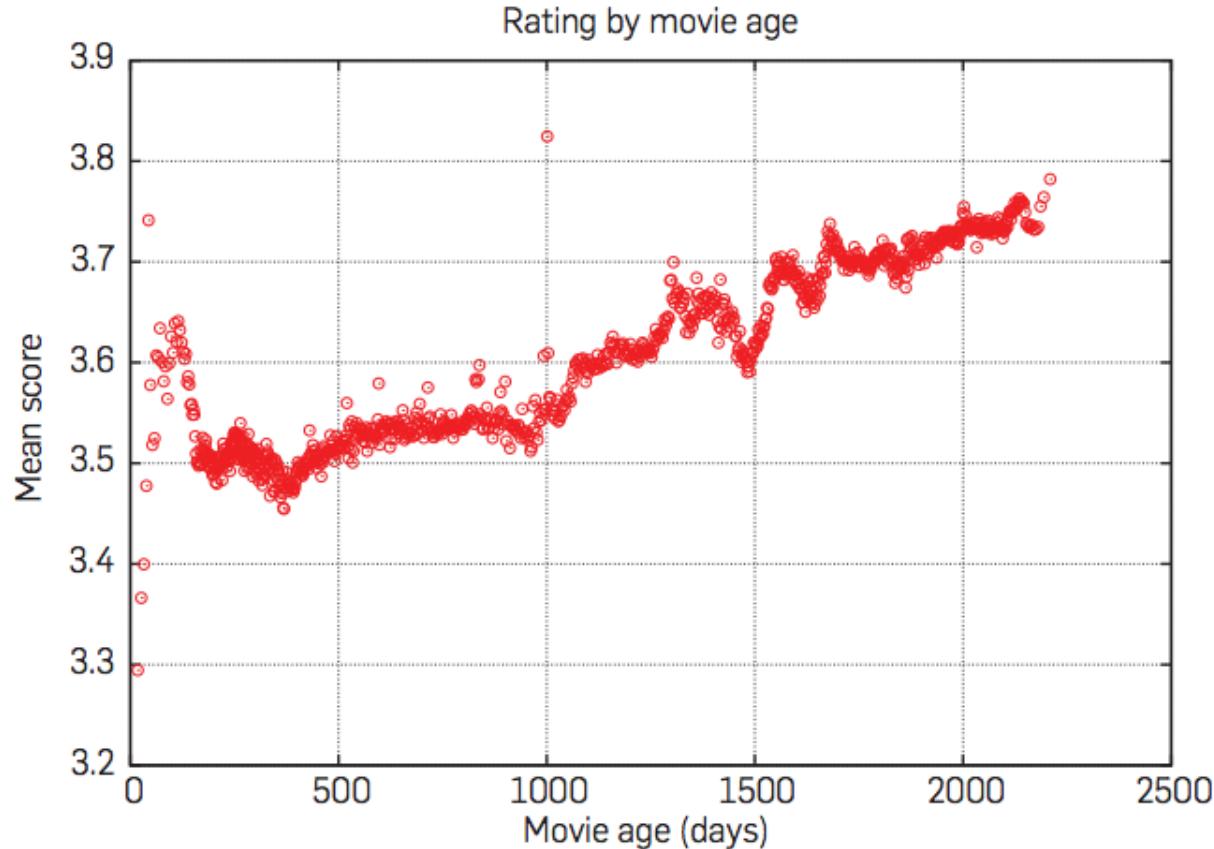


Yehuda Koren, 2009

Time matters (Koren, 2009)

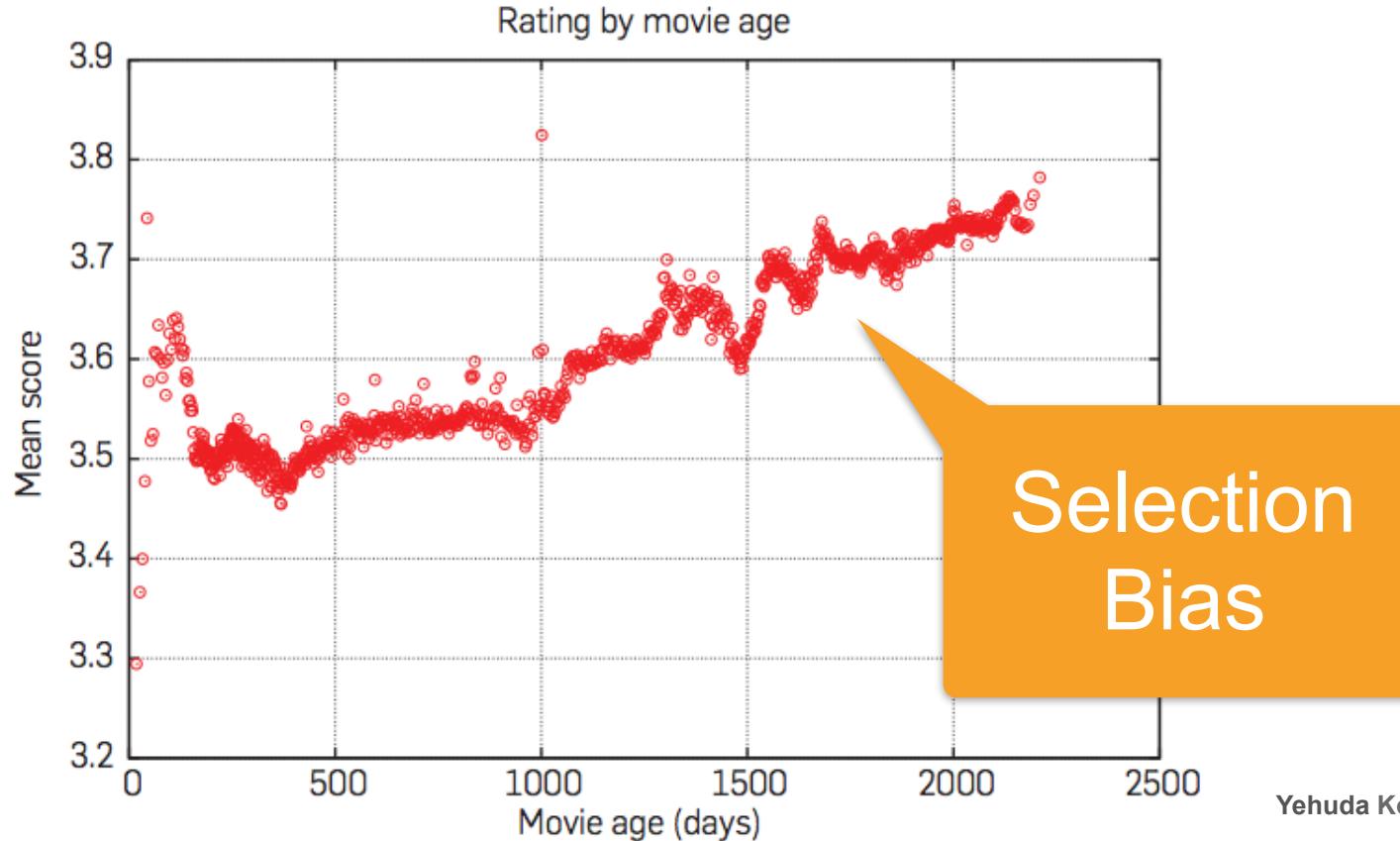


Time matters (Koren, 2009)

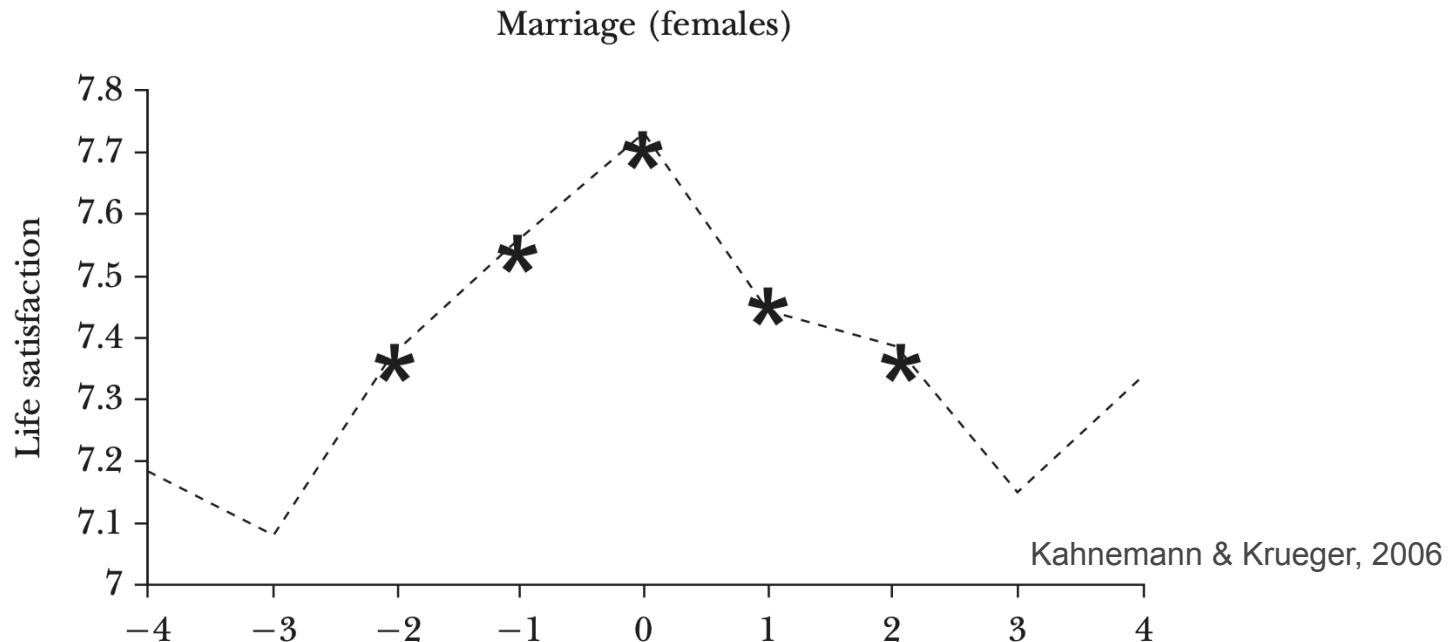


Yehuda Koren, 2009

Time matters (Koren, 2009)



Average Life Satisfaction for a Sample of German Women (by year of marriage $t = 0$)

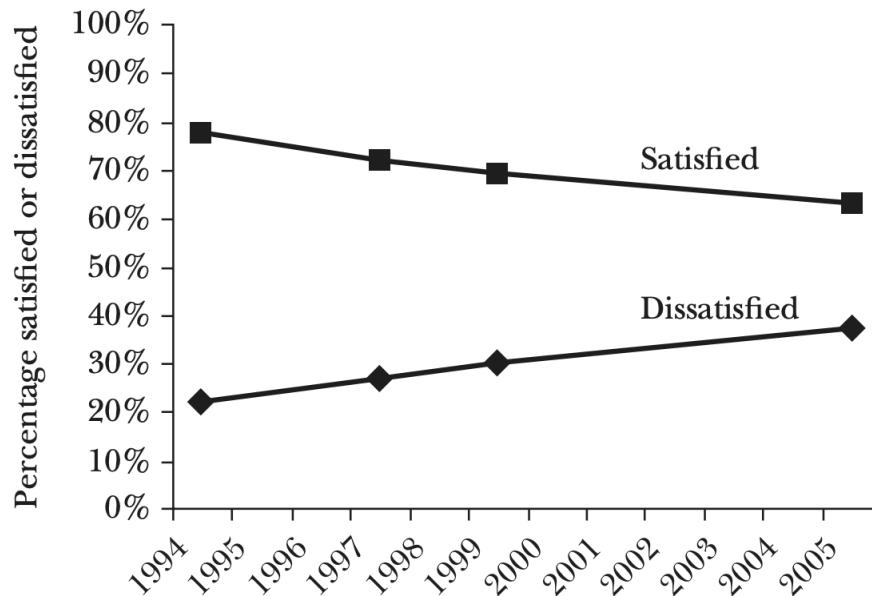


Source: Clark, Diener, Georgellis and Lucas (2003), using data from the German Socioeconomic Panel.

Note: An asterisk indicates that life satisfaction is significantly different from the baseline level.

Life Satisfaction in China as Average Real Income Rises by 250 Percent

Overall, how satisfied or dissatisfied are you with the way things are going in your life today? Would you say you are very satisfied, somewhat satisfied, somewhat dissatisfied, or very dissatisfied?



Kahnemann & Krueger, 2006

Source: Derived from Richard Burkholder, "Chinese Far Wealthier Than a Decade Ago—but Are They Happier?" The Gallup Organization, <<http://www.gallup.com/poll/content/login.aspx?ci=14548>>.

numpy.d2l.ai



TL;DR - Data usually isn't IID

inventory

Christmas

Black Friday

prime day

back to school

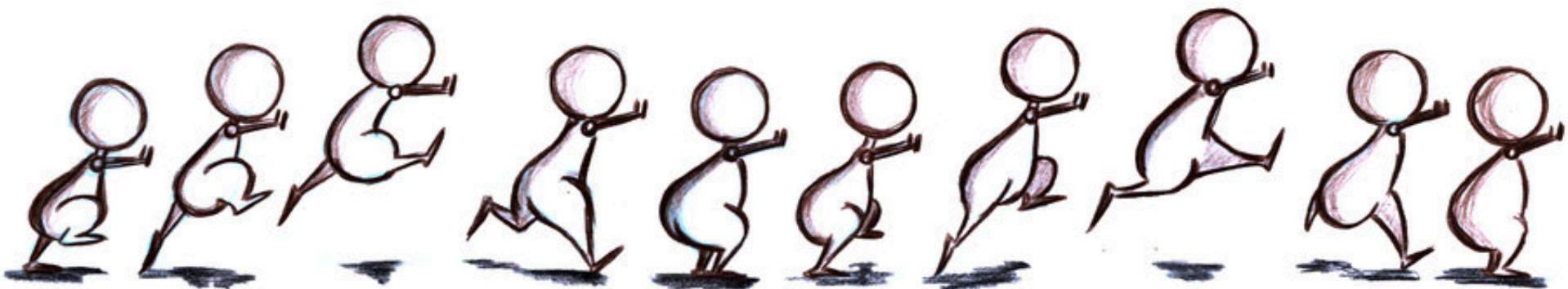
Q2 earnings

rate cuts

orange hair tweets

rating agencies

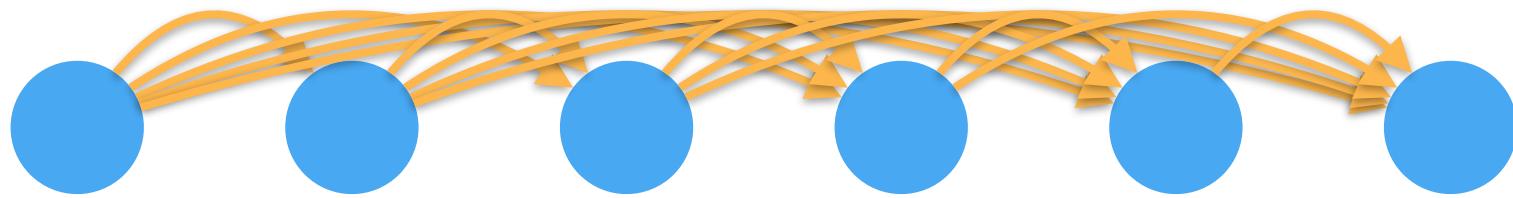
Sequence Models



Sequence Model

$$(x_1, \dots, x_T) \sim p(x)$$

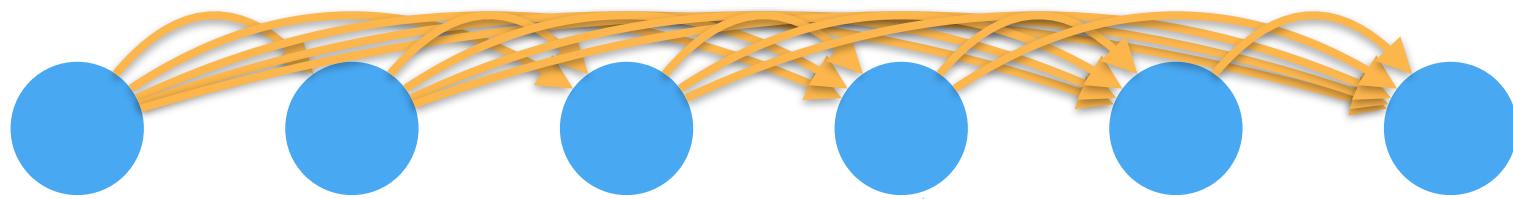
$$p(x) = p(x_1) \cdot p(x_2 | x_1) \cdot p(x_3 | x_1, x_2) \cdot \dots \cdot p(x_T | x_1, \dots, x_{T-1})$$



Sequence Model

$$(x_1, \dots, x_T) \sim p(x)$$

$$p(x) = p(x_1) \cdot p(x_2 | x_1) \cdot p(x_3 | x_1, x_2) \cdot \dots \cdot p(x_T | x_1, \dots, x_{T-1})$$



- Autoregressive model

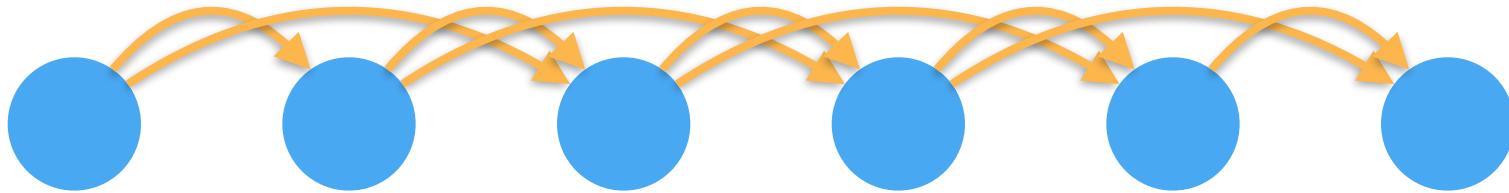
Exponentially expensive

$$p(x_t | x_1, \dots, x_{t-1}) = p(x_t | f(x_1, \dots, x_{t-1}))$$

Some function of
previously seen data

Plan A - Markov Assumption

$$p(x) = p(x_1) \cdot p(x_2 | x_1) \cdot p(x_3 | x_1, x_2) \cdot \dots \cdot p(x_T | x_{T-\tau}, \dots, x_{T-1})$$



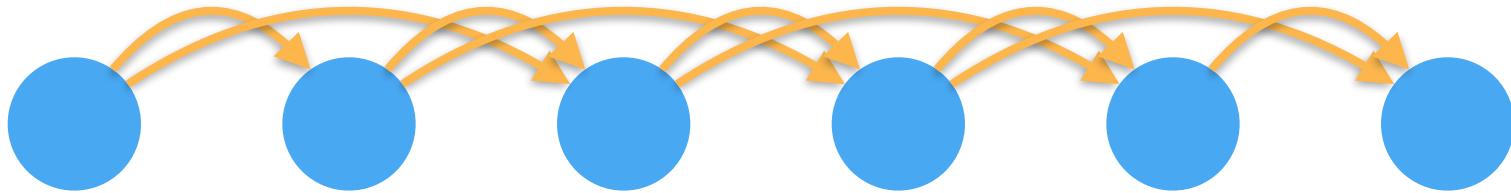
- Assume that only a few steps in the past matter
- Autoregressive model

$$p(x_t | x_1, \dots, x_{t-1}) = p(x_t | f(x_{t-\tau}, \dots, x_{t-1}))$$

Some function of
previously seen data

Plan A - Markov Assumption

$$p(x) = p(x_1) \cdot p(x_2 | x_1) \cdot p(x_3 | x_1, x_2) \cdot \dots \cdot p(x_T | x_{T-\tau}, \dots, x_{T-1})$$



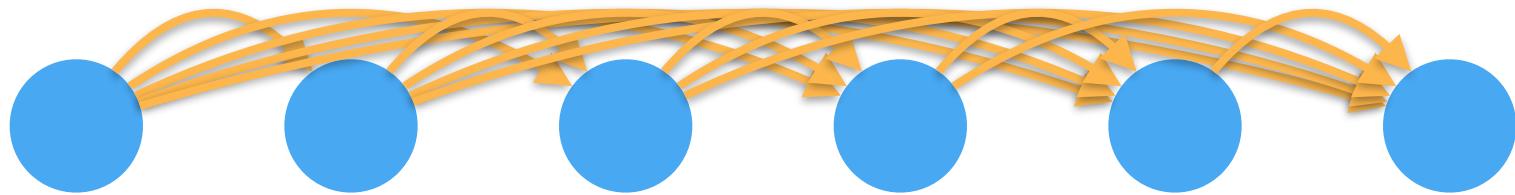
- In practice solve regression problem

$$\hat{x}_t = f(x_{t-\tau}, \dots, x_{t-1})$$

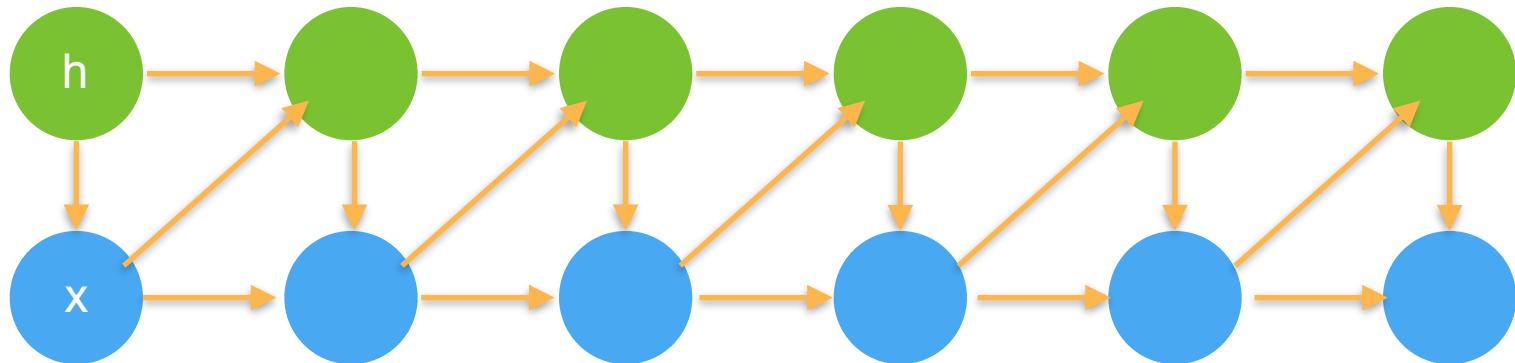
e.g. train an MLP on
previously seen data

Plan B - Latent Variable Model

$$p(x) = p(x_1) \cdot p(x_2 | x_1) \cdot p(x_3 | x_1, x_2) \cdot \dots p(x_T | x_1, \dots x_{T-1})$$



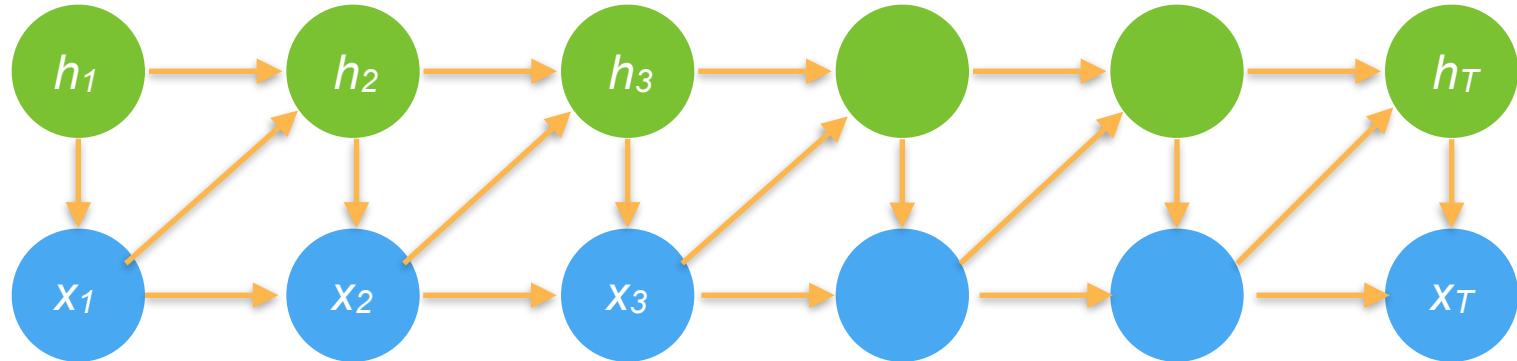
$p(h_t | h_{t-1}, x_{t-1})$ and $p(x_t | h_t, x_{t-1})$



Plan B - Latent Variable Model

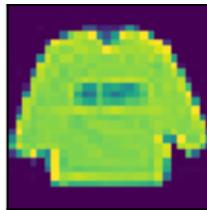
- Latent state, h , summarizes all the relevant information about the past. So we get $h_t = f(x_1, \dots, x_{t-1}) = f(h_{t-1}, x_{t-1})$

$p(h_t | h_{t-1}, x_{t-1})$ and $p(x_t | h_t, x_{t-1})$



Text Preprocessing

- Sequence data has **long dependency** (very costly)
- Truncate into shorter fragments
- Transform examples into mini-batches with ndarrays



→ (batch size, width, height, channel)

The Time Traveller (for so it will be called here) was expounding a recondite matter to us. He twinkled, and his usually pale face was flushed; the fire burned brightly, and the soft radiance reflected from the lights in the lilies of silver caught our glasses. Our chairs, being caressed us rather than submitted to by the luxurious after-dinner atmosphere when free of the trammels of precision. And

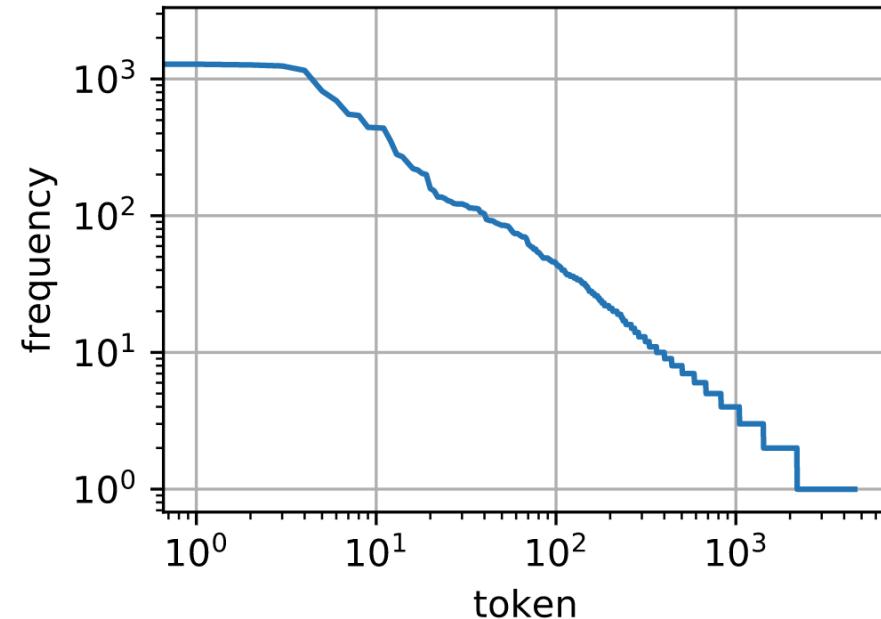
→ (batch size, sentence length)

Tokenization

- Basic Idea - map text into sequence of tokens
 - “Deep learning is fun” -> [“Deep”, “learning”, “is”, “fun”, “.”]
- **Character Encoding** (each character as a token)
 - Small vocabulary
 - Doesn’t work so well (needs to learn spelling)
- **Word Encoding** (each word as a token)
 - Accurate spelling
 - Doesn’t work so well (huge vocabulary = costly multinomial)
- **Byte Pair Encoding** (Goldilocks zone)
 - Frequent subsequences (like syllables)

Vocabulary

- Find unique tokens, map each one into a numerical index
 - “Deep” : 1, “learning” : 2, “is” : 3, “fun” : 4, “.” : 5
- The frequency of words often obeys a power law distribution
 - Map the tailing tokens, e.g. appears < 5 times, into a special “unknown” token



Minibatch Generation

The Time Machine by H. G. Wells

Text Preprocessing Notebook

Language Models

- Tokens not real values (domain is countably finite)

$$p(w_1, w_2, \dots, w_T) = p(w_1) \prod_{t=2}^T p(w_t | w_1, \dots, w_{t-1})$$

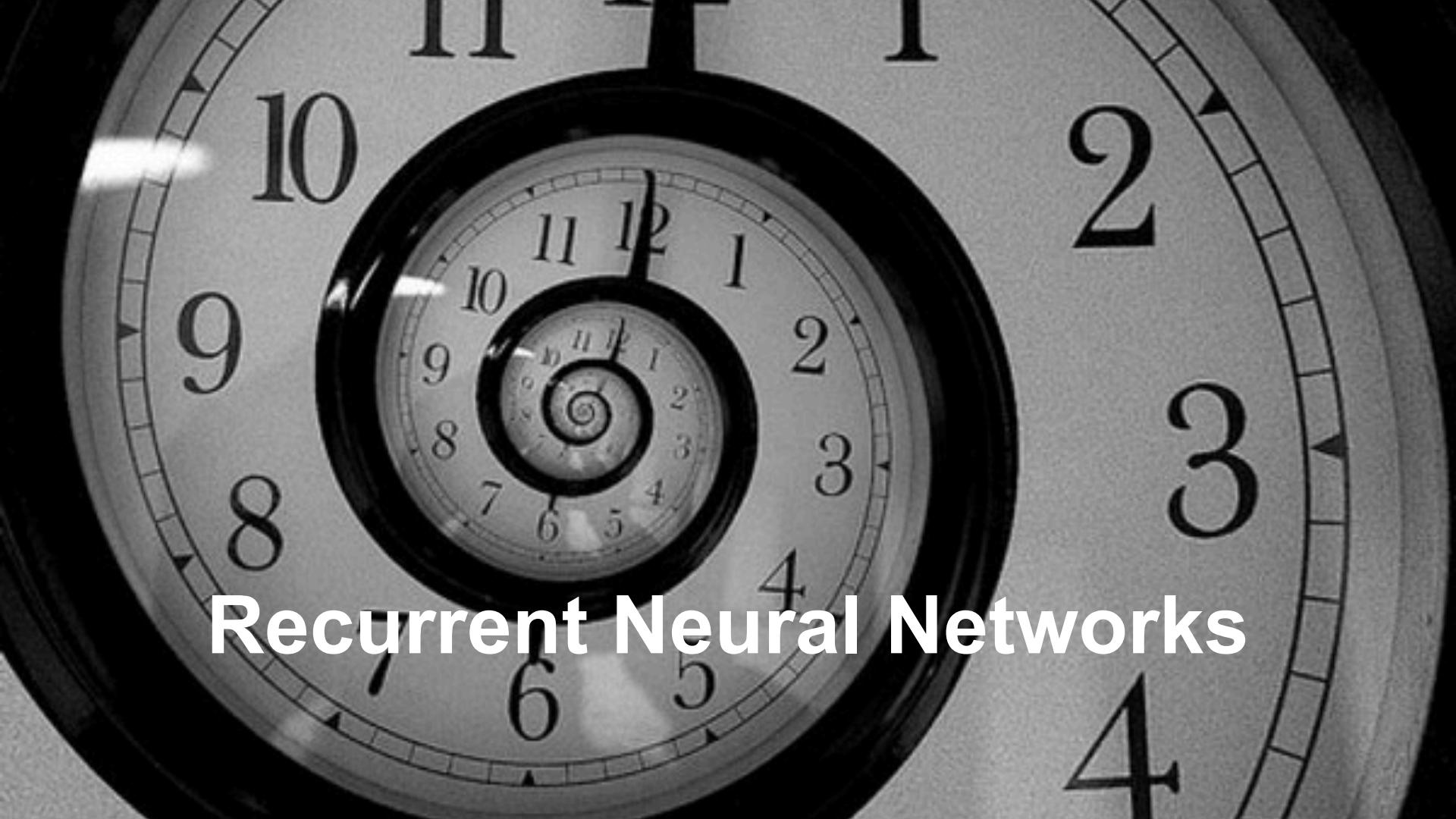
- e.g., $p(\text{deep, learning, is, fun, .})$
 $= p(\text{deep})p(\text{learning} | \text{deep})p(\text{is} | \text{deep, learning})$
 $p(\text{fun} | \text{deep, learning, is})p(\text{.} | \text{deep, learning, is, fun})$
- Estimating it

$$\hat{p}(\text{learning} | \text{deep}) = \frac{n(\text{deep, learning})}{n(\text{deep})}$$

Need Smoothing

Language Modeling

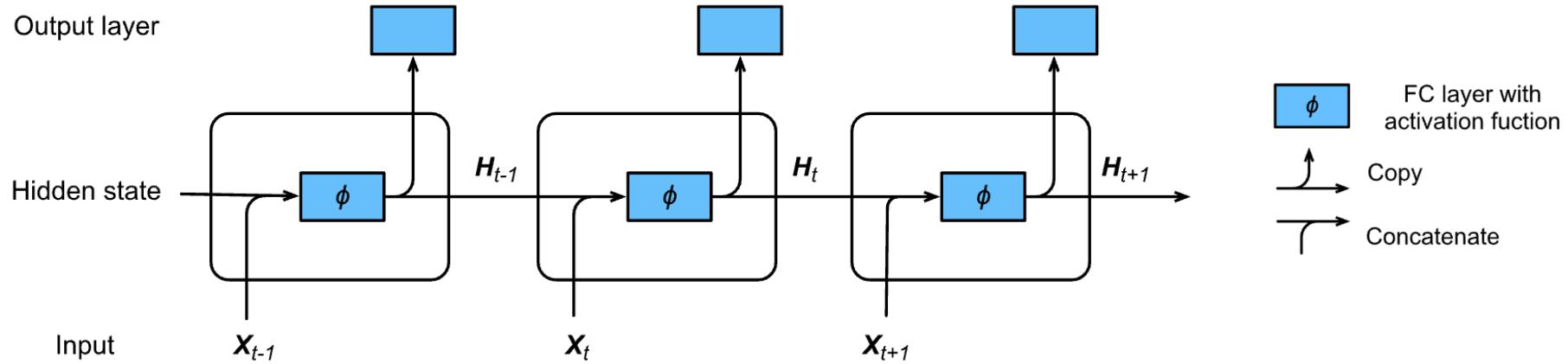
- NLP fundamental tasks
 - Named-entity recognition
 - Part-of-speech tagging
 - Machine translation
 - Question answering
 - Automatic Summarization
 - ...



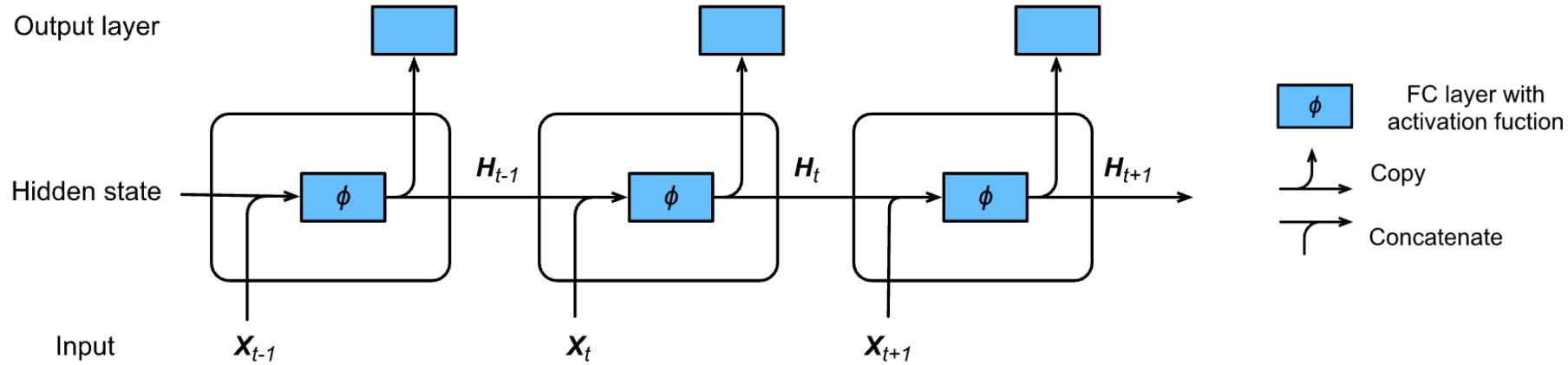
Recurrent Neural Networks

RNN with Hidden States

Output layer



RNN with Hidden States



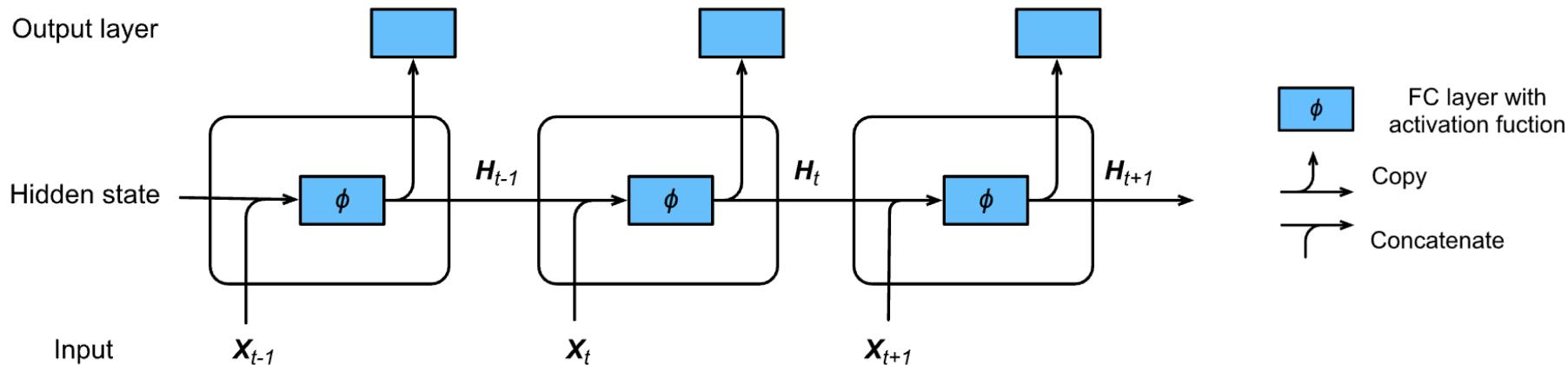
- Hidden State update

$$\mathbf{H}_t = \phi(\mathbf{W}_{hh}\mathbf{H}_{t-1} + \mathbf{W}_{hx}\mathbf{X}_{t-1} + \mathbf{b}_h)$$

- Observation update

$$\mathbf{o}_t = \mathbf{W}_{ho}\mathbf{H}_t + \mathbf{b}_o$$

RNN with Hidden States



- Hidden State update

$$\mathbf{H}_t = \phi(\mathbf{W}_{hh}\mathbf{H}_{t-1} + \mathbf{W}_{hx}\mathbf{X}_{t-1} + \mathbf{b}_h)$$

- Observation update

$$\mathbf{o}_t = \mathbf{W}_{ho}\mathbf{H}_t + \mathbf{b}_o$$

- 2-layer MLP

$$\mathbf{H}_t = \phi(\mathbf{W}_{hx}\mathbf{X}_{t-1} + \mathbf{b}_h)$$

$$\mathbf{o}_t = \mathbf{W}_{ho}\mathbf{H}_t + \mathbf{b}_o$$

Next word prediction

step

1

2

3

4

5

output

output

state

hidden

state

input

The

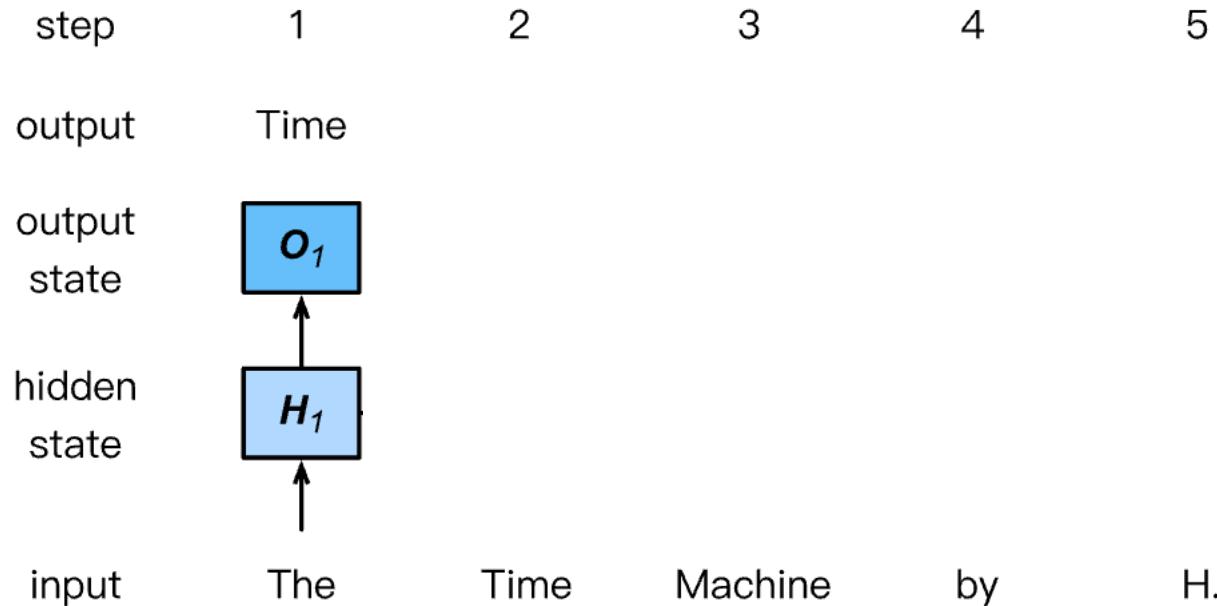
Time

Machine

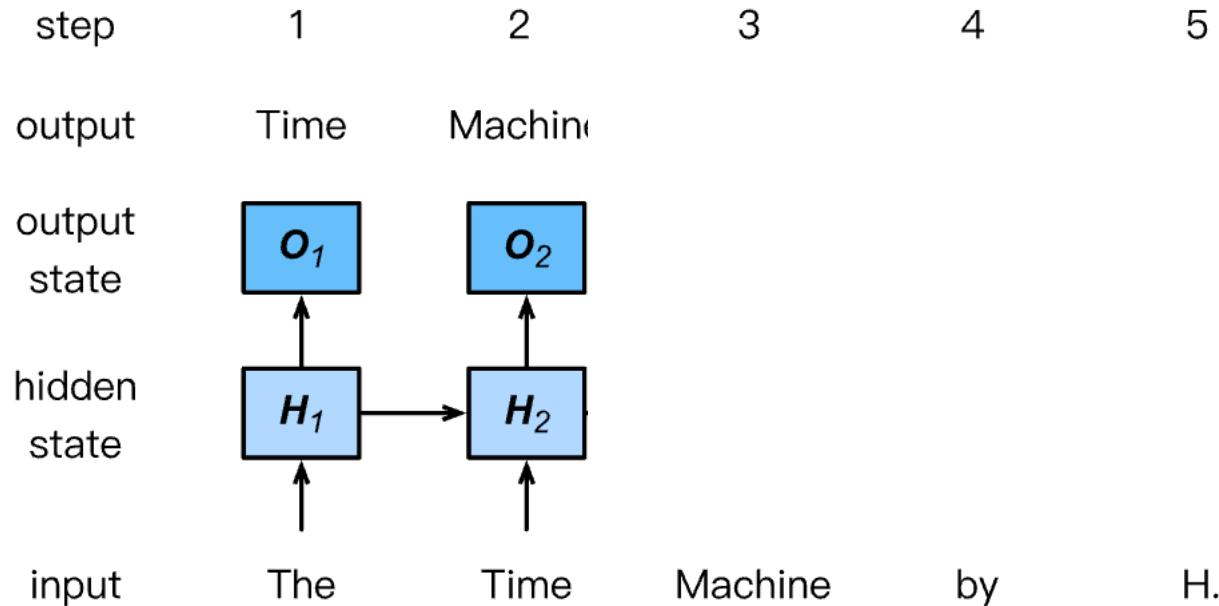
by

H.

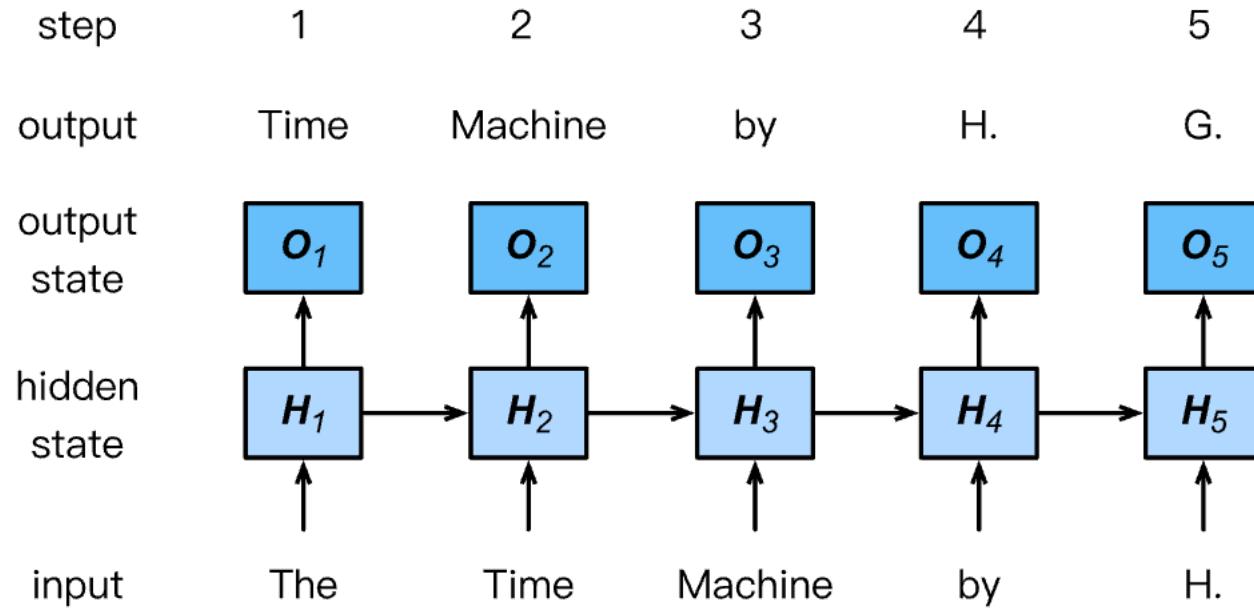
Next word prediction



Next word prediction



Next word prediction



Input Encoding

- Need to map input numerical indices to vectors
 - Pick granularity (words, characters, subwords)
 - Map to indicator vectors

```
npx.one_hot(np.array([0, 2]), len(vocab))
```

```
array([[1., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
       0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
       [0., 0., 1., 0., 0., 0., 0., 0., 0., 0.,
       0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
```

RNN with hidden state mechanics

- Input: vector sequence $\mathbf{x}_1, \dots, \mathbf{x}_T \in \mathbb{R}^d$
- Hidden states: $\mathbf{h}_1, \dots, \mathbf{h}_T \in \mathbb{R}^h$ where $\mathbf{h}_t = f(\mathbf{h}_{t-1}, \mathbf{x}_t)$
- Output: vector sequence $\mathbf{o}_1, \dots, \mathbf{o}_T \in \mathbb{R}^p$ where $\mathbf{o}_t = g(\mathbf{h}_t)$
 - p is the vocabulary size
 - $\mathbf{o}_{t,j}$ is confident score that the t -th timestamp in the sequence equals to j -th token in the vocabulary
- Loss: measure the classification error on T tokens

Gradient Clipping

- Long chain of dependencies for backprop
 - Need to keep a lot of intermediate values in memory
 - Butterfly effect style dependencies
 - Gradients can vanish or explode
- Clipping to prevent divergence

$$\mathbf{g} \leftarrow \min \left(1, \frac{\theta}{\|\mathbf{g}\|} \right) \mathbf{g}$$

rescales to gradient of size at most θ

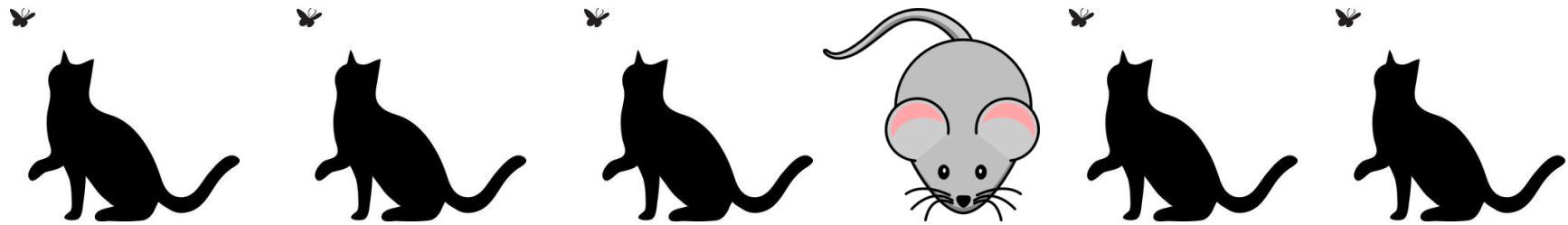
RNN Notebook

Gated Recurrent Unit (GRU)



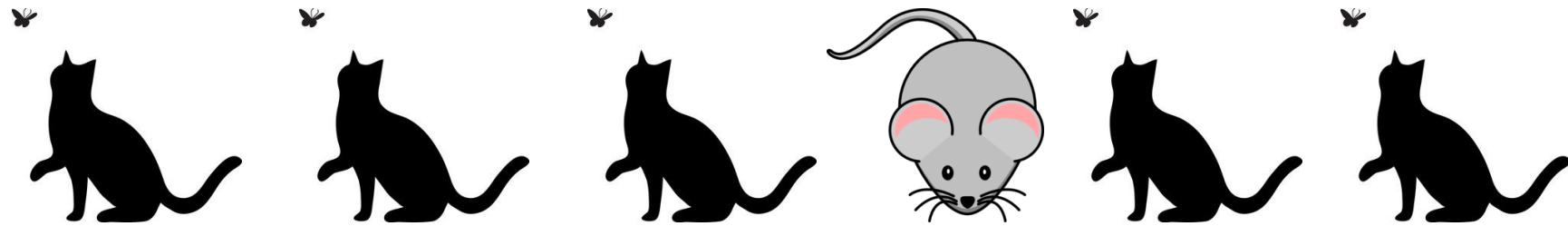
Paying attention to a sequence

- Not all observations are equally relevant



Paying attention to a sequence

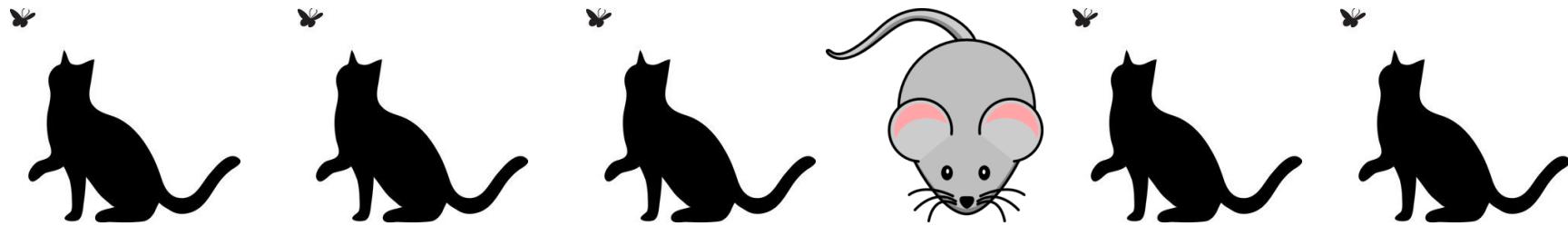
- Not all observations are equally relevant



- Need mechanism to **pay attention (update gate)**
e.g., an early observation is highly significant for predicting all future observations. We would like to have some mechanism for **storing/updaing** vital early information in a memory cell.

Paying attention to a sequence

- Not all observations are equally relevant



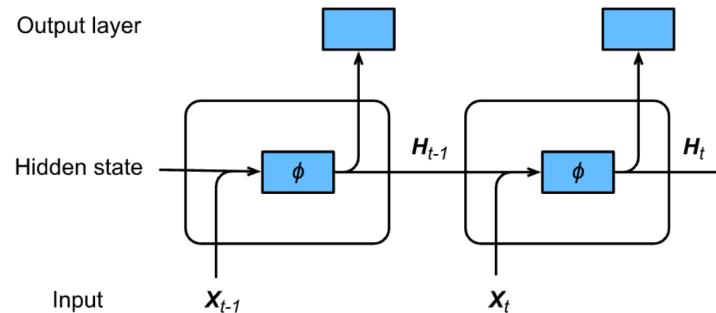
- Need mechanism to **forget (reset gate)**

e.g., There is a logical break between parts of a sequence. For instance there might be a transition between chapters in a book, a transition between a bear and a bull market for securities, etc.

From RNN to GRU

$$\mathbf{H}_t = \phi(\mathbf{W}_{hh}\mathbf{H}_{t-1} + \mathbf{W}_{hx}\mathbf{X}_{t-1} + \mathbf{b}_h)$$

$$\mathbf{o}_t = \mathbf{W}_{ho}\mathbf{H}_t + \mathbf{b}_o$$



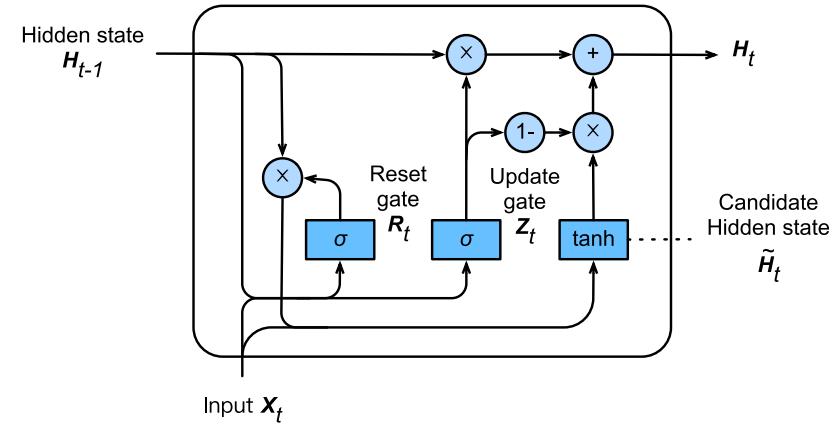
$$\mathbf{R}_t = \sigma(\mathbf{X}_t \mathbf{W}_{xr} + \mathbf{H}_{t-1} \mathbf{W}_{hr} + \mathbf{b}_r),$$

$$\mathbf{Z}_t = \sigma(\mathbf{X}_t \mathbf{W}_{xz} + \mathbf{H}_{t-1} \mathbf{W}_{hz} + \mathbf{b}_z)$$

$$\tilde{\mathbf{H}}_t = \tanh(\mathbf{X}_t \mathbf{W}_{xh} + (\mathbf{R}_t \odot \mathbf{H}_{t-1}) \mathbf{W}_{hh} + \mathbf{b}_h)$$

$$\mathbf{H}_t = \mathbf{Z}_t \odot \mathbf{H}_{t-1} + (1 - \mathbf{Z}_t) \odot \tilde{\mathbf{H}}_t$$

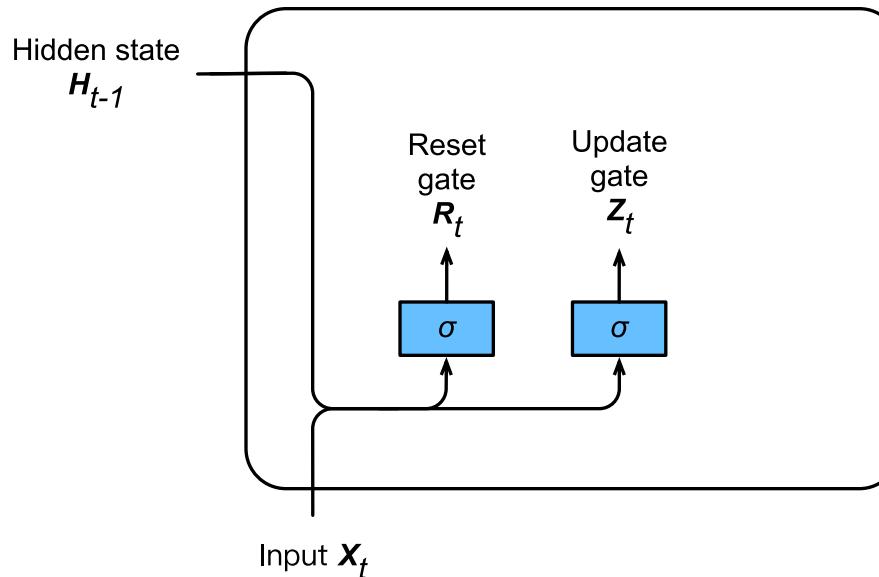
$$\mathbf{o}_t = \mathbf{W}_{ho}\mathbf{H}_t + \mathbf{b}_o$$



GRU - Gates

$$R_t = \sigma(X_t W_{xr} + H_{t-1} W_{hr} + b_r),$$

$$Z_t = \sigma(X_t W_{xz} + H_{t-1} W_{hz} + b_z)$$



FC layer with
activation function



Element-wise
Operator



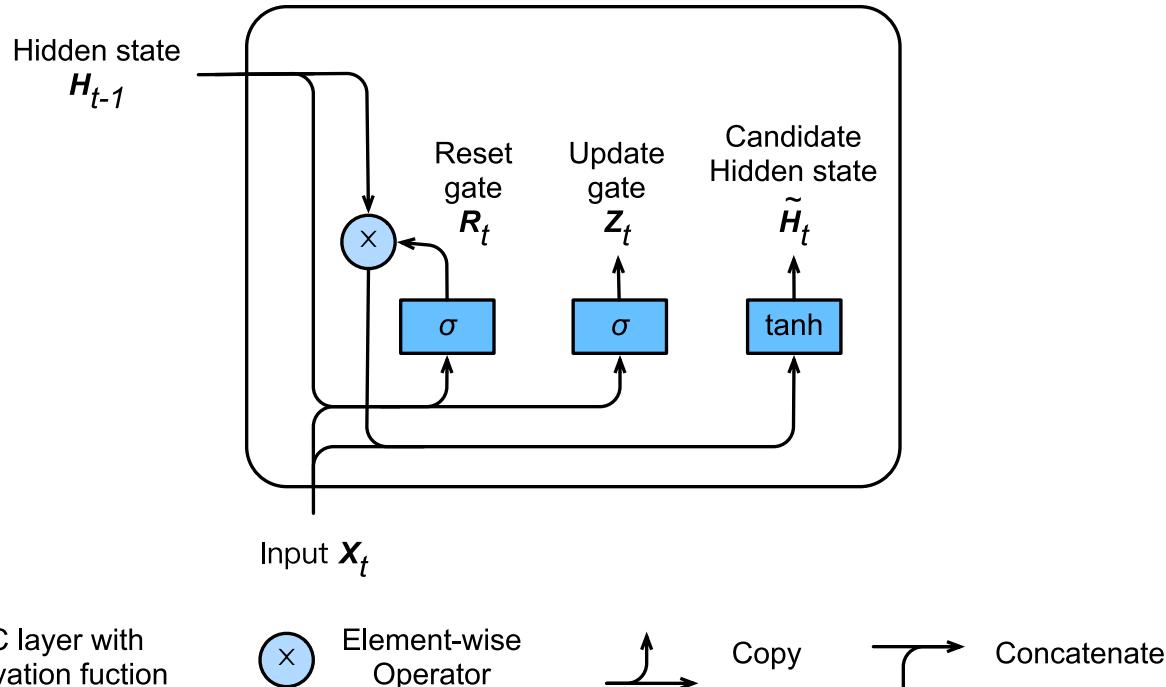
Copy



Concatenate

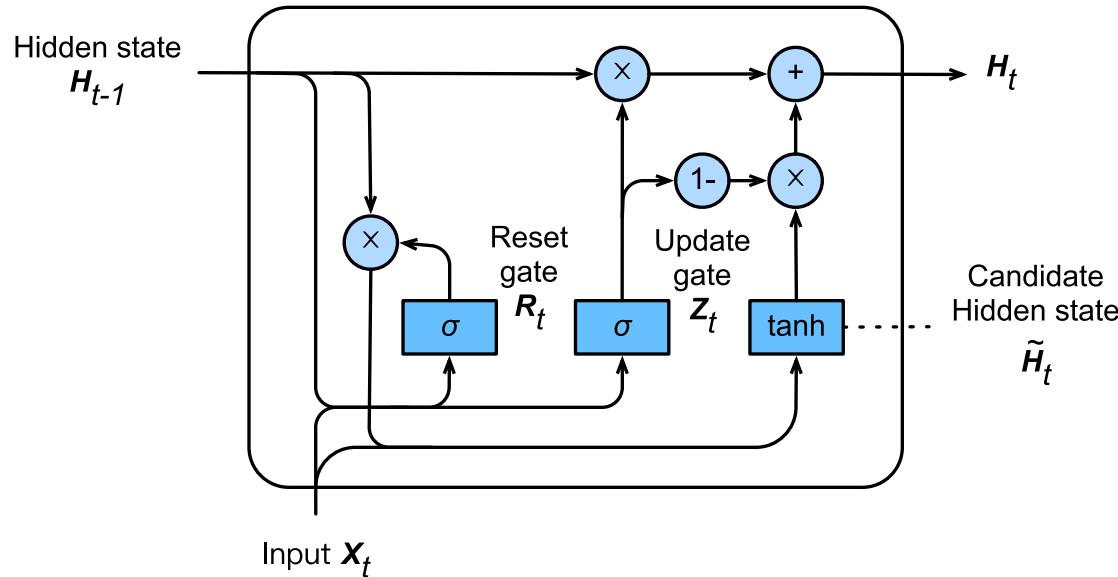
GRU - Candidate Hidden State

$$\tilde{H}_t = \tanh(X_t W_{xh} + (R_t \odot H_{t-1}) W_{hh} + b_h)$$



Hidden State

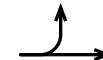
$$H_t = Z_t \odot H_{t-1} + (1 - Z_t) \odot \tilde{H}_t$$



FC layer with
activation function



Element-wise
Operator



Copy



Concatenate

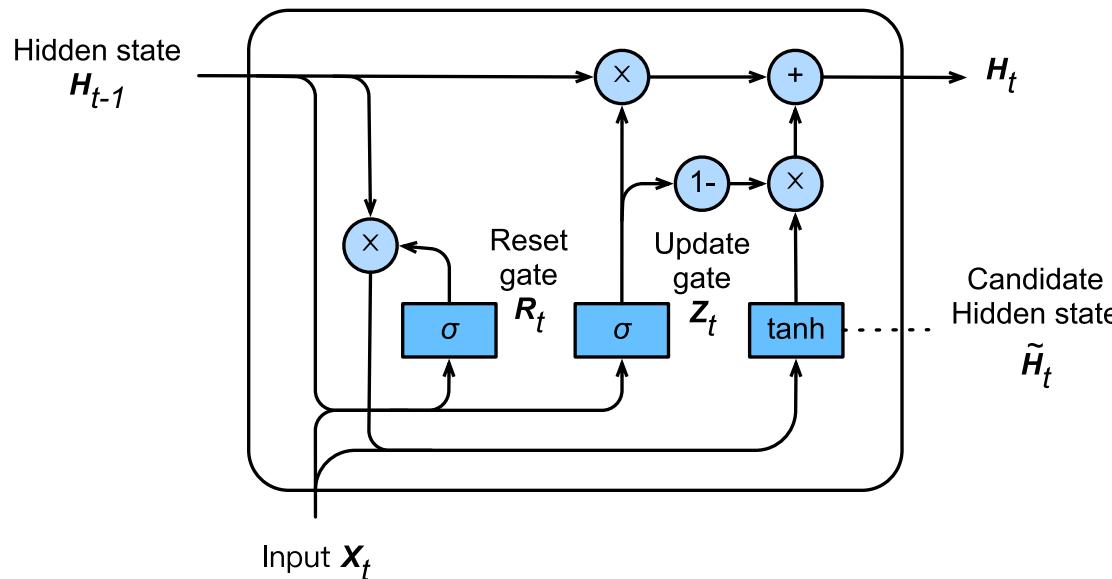
Summary

$$R_t = \sigma(X_t W_{xr} + H_{t-1} W_{hr} + b_r),$$

$$Z_t = \sigma(X_t W_{xz} + H_{t-1} W_{hz} + b_z)$$

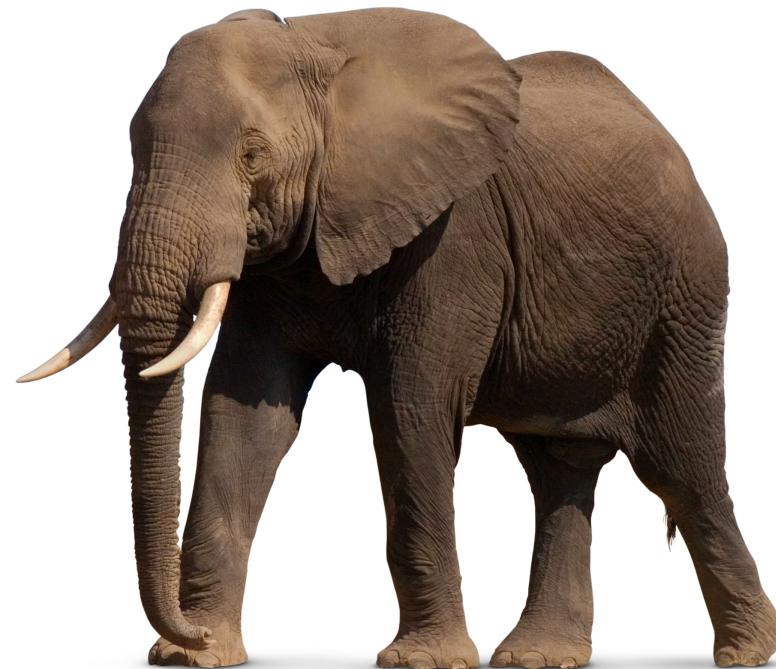
$$\tilde{H}_t = \tanh(X_t W_{xh} + (R_t \odot H_{t-1}) W_{hh} + b_h)$$

$$H_t = Z_t \odot H_{t-1} + (1 - Z_t) \odot \tilde{H}_t$$



GRU Notebook

Long Short Term Memory



GRU and LSTM

$$R_t = \sigma(X_t W_{xr} + H_{t-1} W_{hr} + b_r),$$

$$Z_t = \sigma(X_t W_{xz} + H_{t-1} W_{hz} + b_z)$$

$$\tilde{H}_t = \tanh(X_t W_{xh} + [R_t] \odot H_{t-1}) W_{hh} + b_h)$$

$$H_t = [Z_t] \odot H_{t-1} + (1 - Z_t) \odot \tilde{H}_t$$

$$I_t = \sigma(X_t W_{xi} + H_{t-1} W_{hi} + b_i)$$

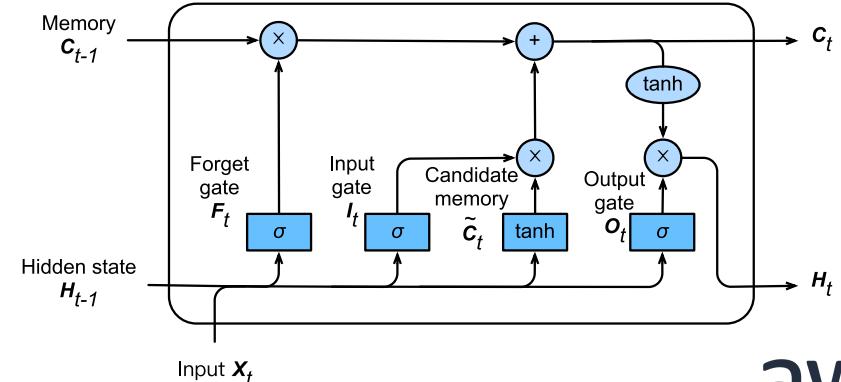
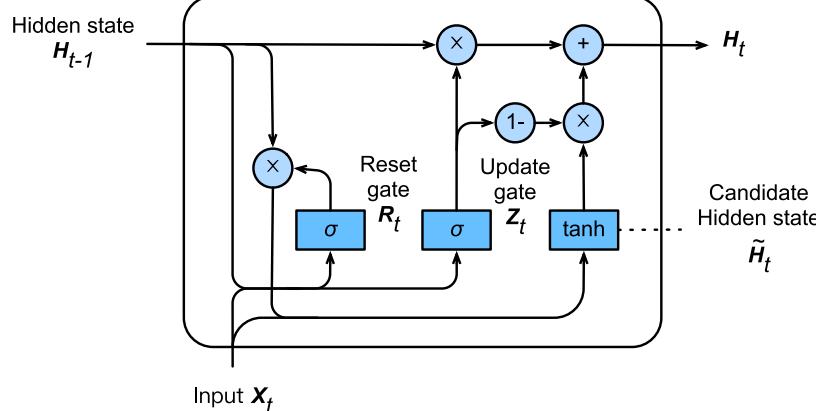
$$F_t = \sigma(X_t W_{xf} + H_{t-1} W_{hf} + b_f)$$

$$O_t = \sigma(X_t W_{xo} + H_{t-1} W_{ho} + b_o)$$

$$\tilde{C}_t = \tanh(X_t W_{xc} + H_{t-1} W_{hc} + b_c)$$

$$C_t = [F_t] \odot C_{t-1} + [I_t] \odot \tilde{C}_t$$

$$H_t = [O_t] \odot \tanh(C_t)$$



Long Short Term Memory

- **Forget gate**

Reset the memory cell values

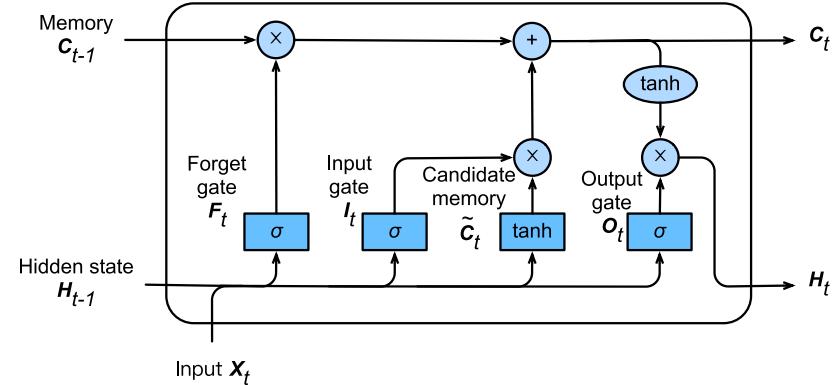
- **Input gate**

Decide whether we should ignore the input data

- **Output gate**

Decide whether the hidden state is used for the output generated by the LSTM

- **Hidden state and Memory cell**

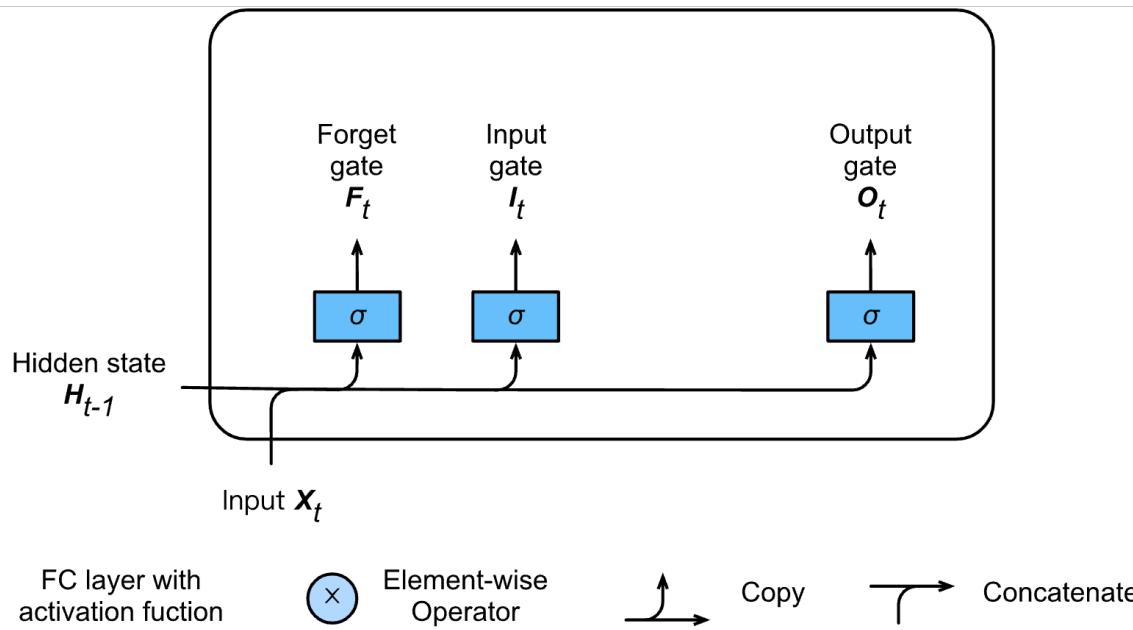


Gates

$$I_t = \sigma(X_t W_{xi} + H_{t-1} W_{hi} + b_i)$$

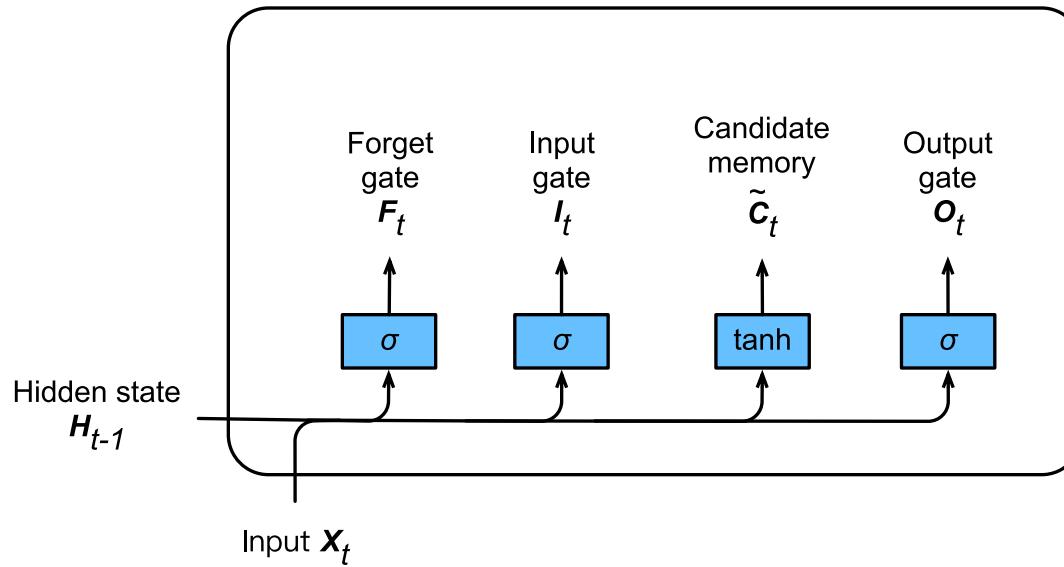
$$F_t = \sigma(X_t W_{xf} + H_{t-1} W_{hf} + b_f)$$

$$O_t = \sigma(X_t W_{xo} + H_{t-1} W_{ho} + b_o)$$



Candidate Memory Cell

$$\tilde{C}_t = \tanh(X_t W_{xc} + H_{t-1} W_{hc} + b_c)$$



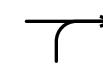
FC layer with
activation function



Element-wise
Operator



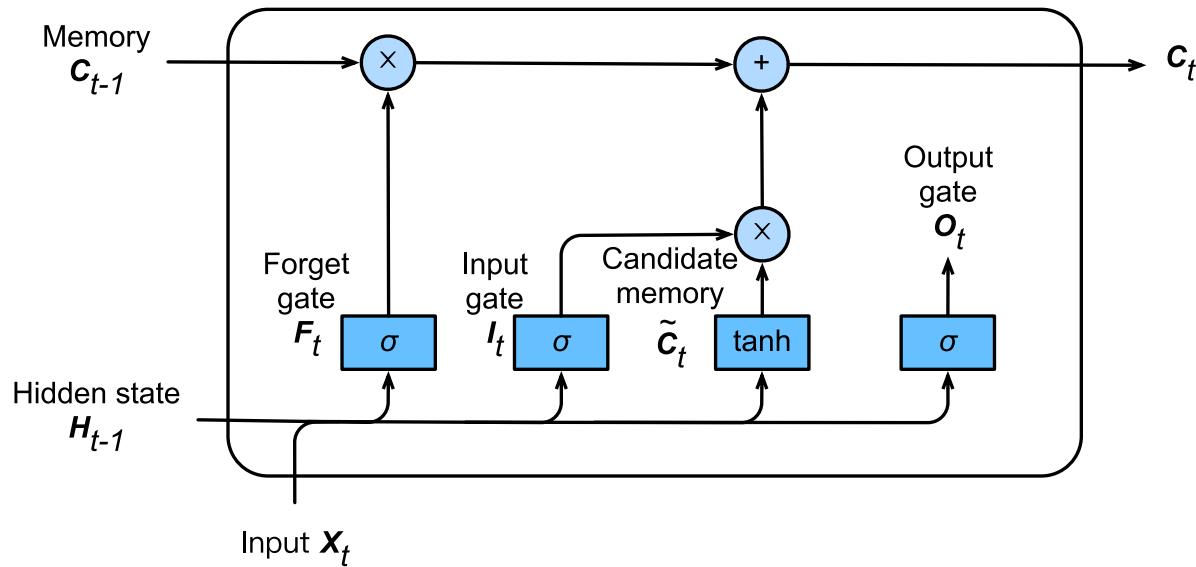
Copy



Concatenate

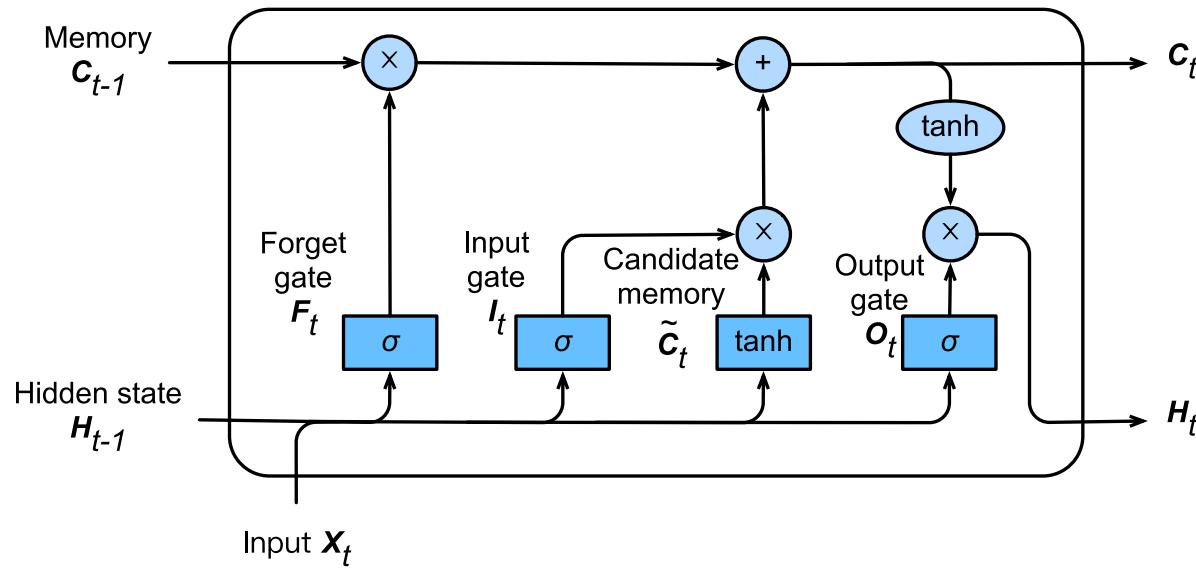
Memory Cell

$$C_t = F_t \odot C_{t-1} + I_t \odot \tilde{C}_t$$

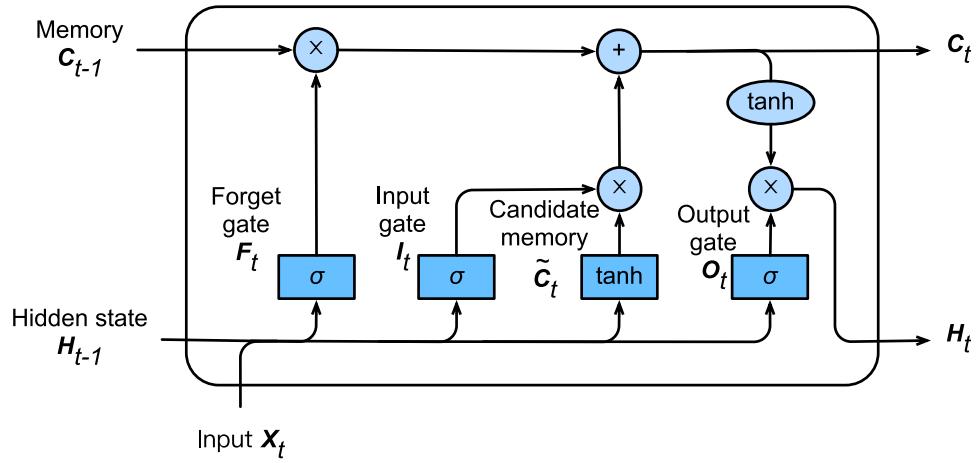


Hidden State / Output

$$H_t = O_t \odot \tanh(C_t)$$



Hidden State / Output



$$I_t = \sigma(X_t W_{xi} + H_{t-1} W_{hi} + b_i)$$

$$F_t = \sigma(X_t W_{xf} + H_{t-1} W_{hf} + b_f)$$

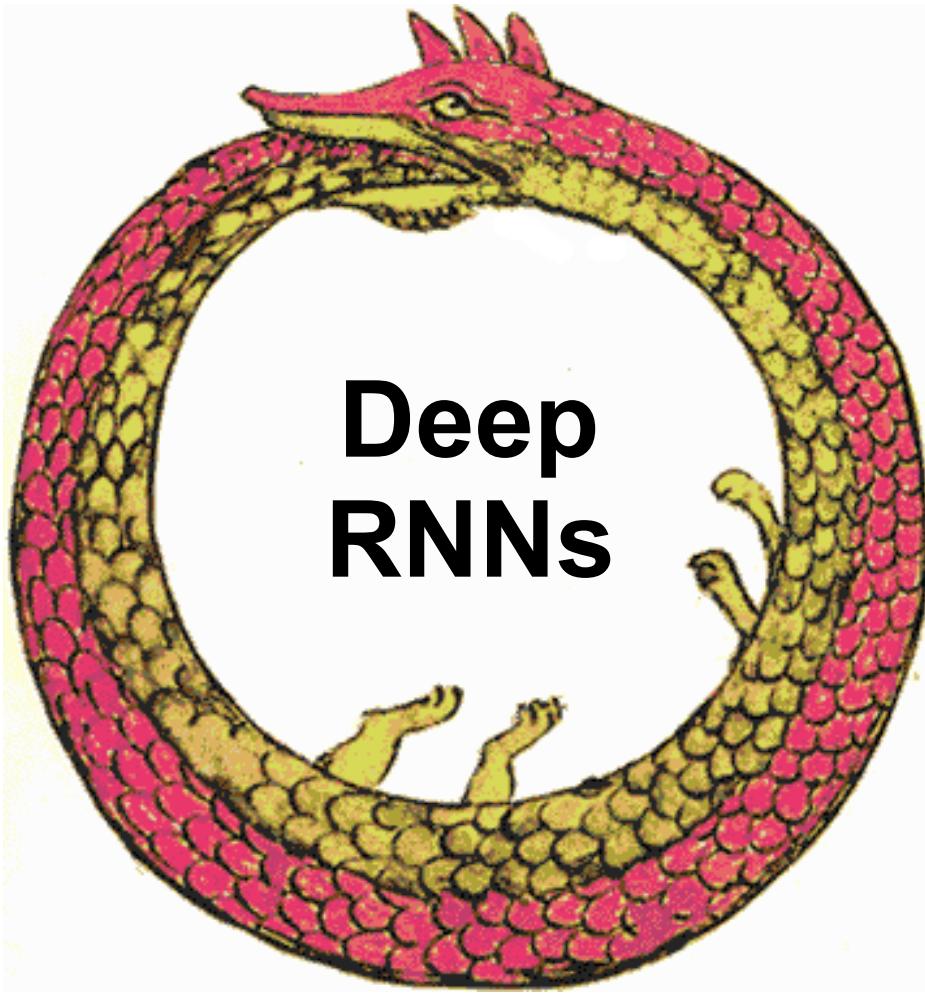
$$O_t = \sigma(X_t W_{xo} + H_{t-1} W_{ho} + b_o)$$

$$\tilde{C}_t = \tanh(X_t W_{xc} + H_{t-1} W_{hc} + b_c)$$

$$C_t = F_t \odot C_{t-1} + I_t \odot \tilde{C}_t$$

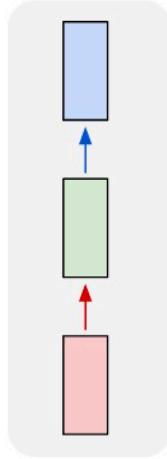
$$H_t = O_t \odot \tanh(C_t)$$

LSTM Notebook

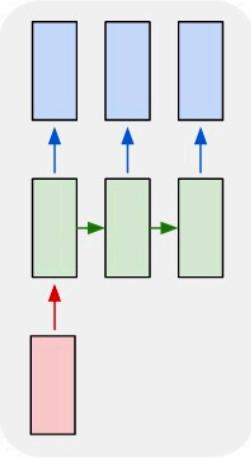


Using RNNs

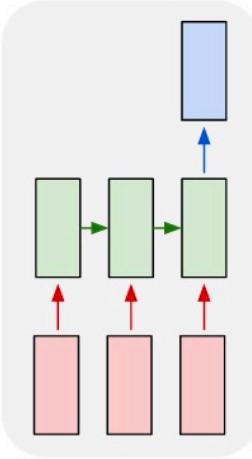
one to one



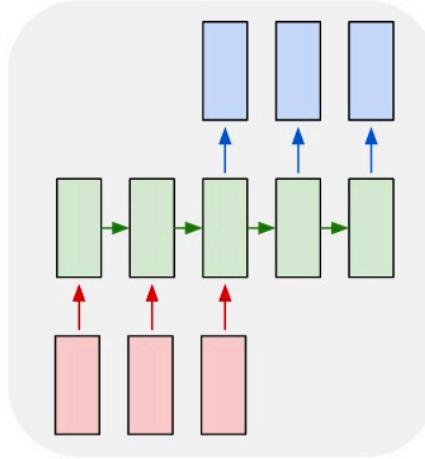
one to many



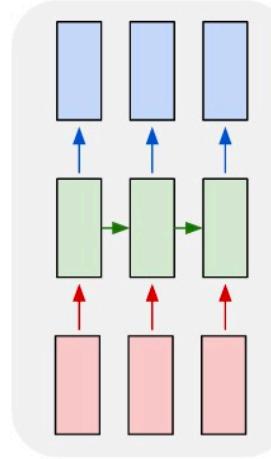
many to one



many to many



many to many



Poetry
Generation

Sentiment
Analysis

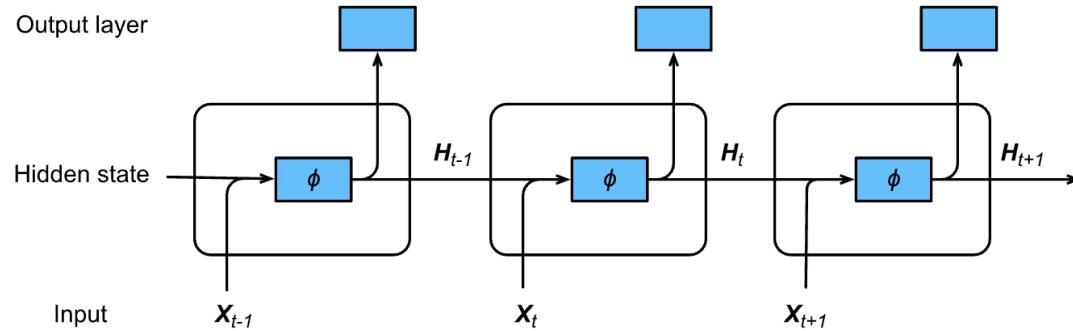
Document
Classification

Question
Answering

Machine
Translation

Named
Entity
Tagging

Recall - RNNs Architecture



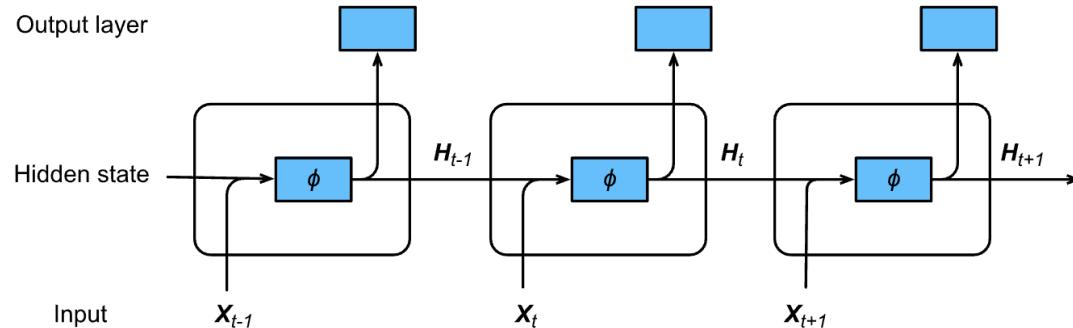
- Hidden State update

$$\mathbf{H}_t = \phi(\mathbf{W}_{hh}\mathbf{H}_{t-1} + \mathbf{W}_{hx}\mathbf{X}_{t-1} + \mathbf{b}_h)$$

- Observation update

$$\mathbf{o}_t = \mathbf{W}_{ho}\mathbf{H}_t + \mathbf{b}_o$$

Recall - RNNs Architecture



- Hidden State update

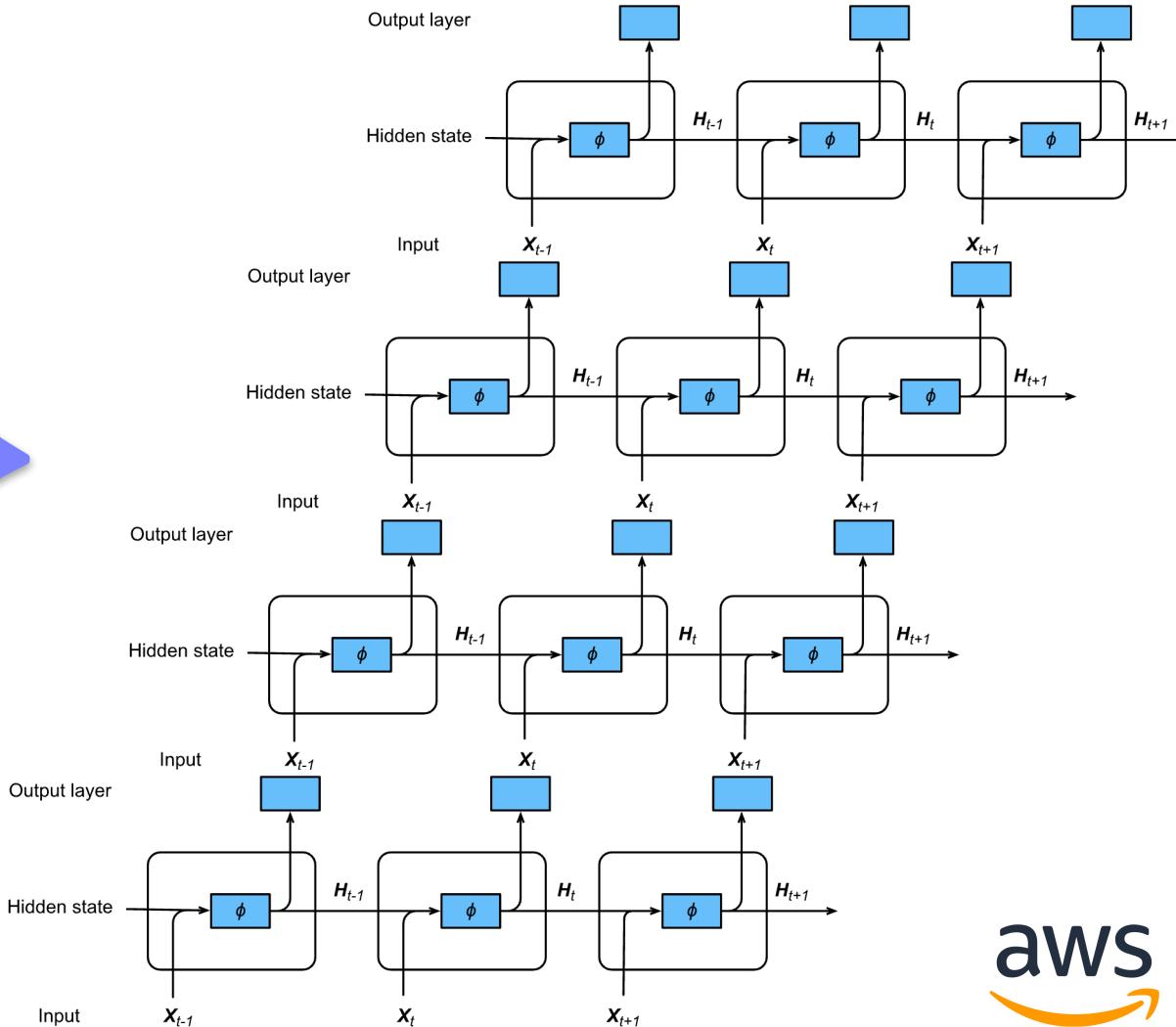
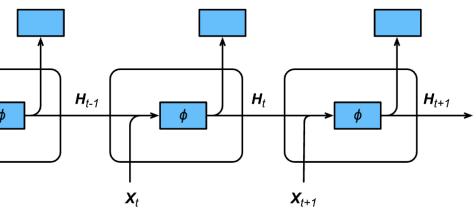
$$\mathbf{H}_t = \phi(\mathbf{W}_{hh}\mathbf{H}_{t-1} + \mathbf{W}_{hx}\mathbf{X}_{t-1} + \mathbf{b}_h)$$

- Observation update

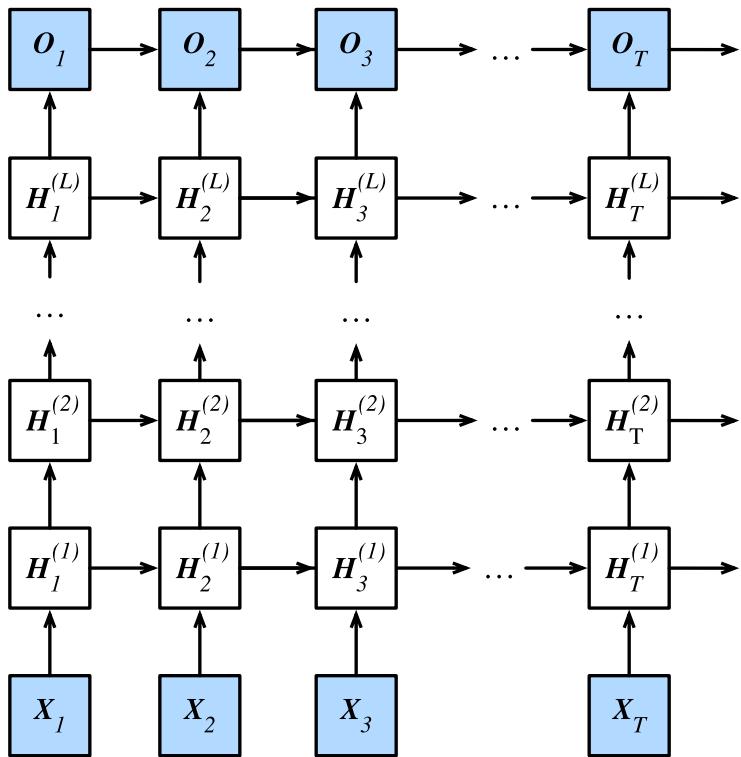
$$\mathbf{o}_t = \mathbf{W}_{ho}\mathbf{H}_t + \mathbf{b}_o$$

How to make
more nonlinear?

We go deeper



We go deeper



- Shallow RNN
 - Input
 - Hidden layer
 - Output
- Deep RNN
 - Input
 - **Hidden layer**
 - **Hidden layer**
 - ...
 - Output

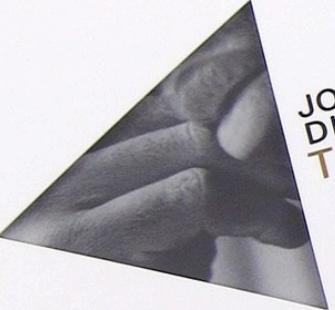
$$\mathbf{H}_t = f(\mathbf{H}_{t-1}, \mathbf{X}_t)$$

$$\mathbf{O}_t = g(\mathbf{H}_t)$$

$$\mathbf{H}_t^1 = f_1(\mathbf{H}_{t-1}^1, \mathbf{X}_t)$$

$$\mathbf{H}_t^j = f_j(\mathbf{H}_{t-1}^j, \mathbf{H}_t^{j-1})$$

$$\mathbf{O}_t = g(\mathbf{H}_t^L)$$



JOHN COLTRANE BOTH
DIRECTIONS AT ONCE
THE LOST ALBUM

Bidirectional RNNS



I am _____
I am _____ very hungry,
I am _____ very hungry, I could eat half a pig.

I am **hungry**.

I am **not** very hungry,

I am **very** very hungry, I could eat half a pig.

The Future Matters

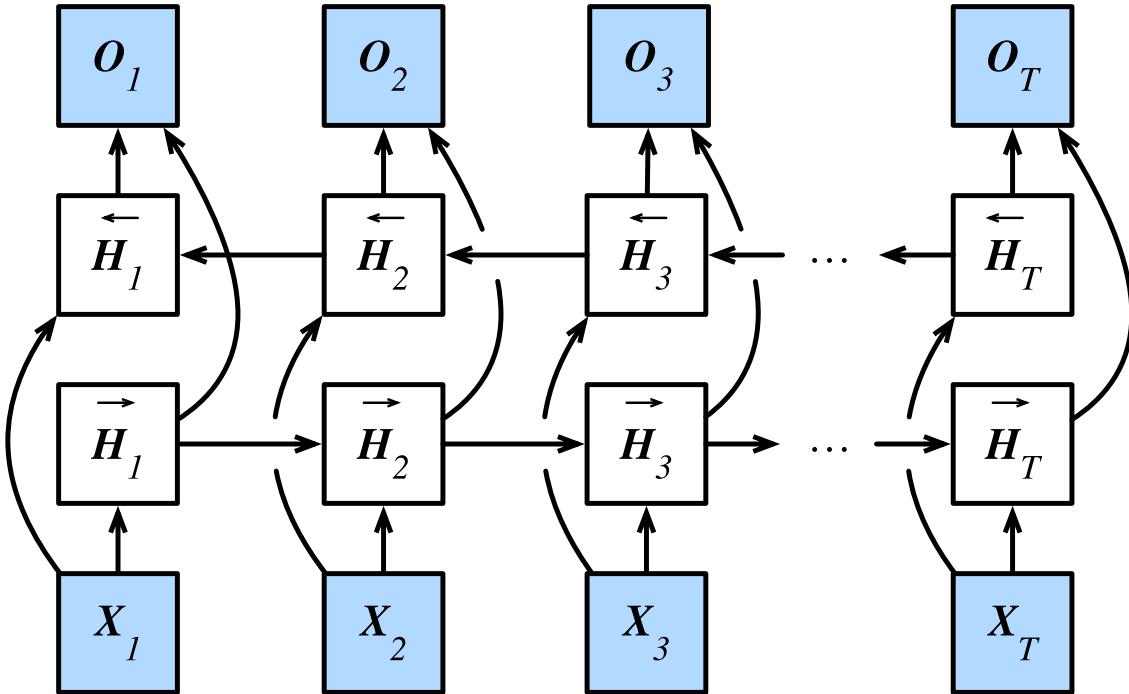
I am **hungry**.

I am **not** very hungry,

I am **very** very hungry, I could eat half a pig.

- Very different words to fill in, depending on past and **future** context of a word.
- RNNs so far only look at the past
- In interpolation (fill in) we can use the future, too.

Bidirectional RNN



- One RNN forward
- Another one backward
- Combine both hidden states for output generation

Summary

- Sequence models and language models
- Recurrent neural networks (RNN)
- GRU and LSTM
- Deep RNNs, Bi-RNNs

More Contents in D2L.ai

Math

- Linear algebra, Prob, Calculus & Statistics Gradient

Machine learning

- Loss function
- Regularization
- Model selection
- Environment

Optimization

- Convex Optimization
- Momentum, RMSProp, Adam

Attention

- Seq2seq w/ attention
- Transformer
- BERT

numpy.d2l.ai

Basic

- NDarray
- Autograd
- Gluon

Basic models

- Linear regression
- Image classification
- Softmax regression
- Multilayer perceptron

RNNs and

- Recurrent networks (RNN, GRU, LSTM) for language modeling
- Word embedding
- Seq2seq for machine translation

Performance

- Numerical stability
- Multi-GPU Training

CNN

- Convolution, LeNet
- Alex, VGG, Inception, ResNet

CV

- Data Augmentation
- Fine-tuning
- Object detection
- Segmentation

GAN

- Generative Adversarial Networks
- DCGAN

- What we covered
- Not

