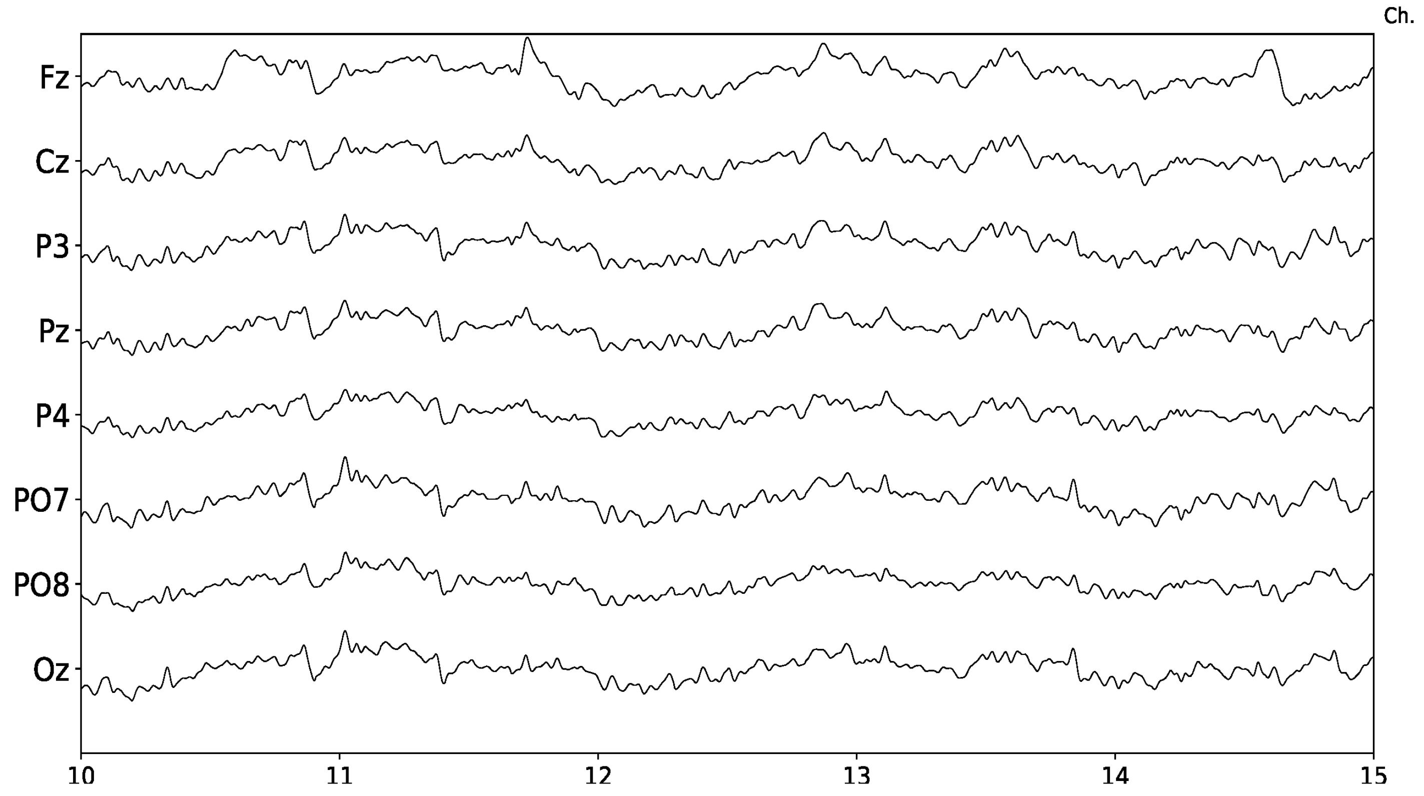
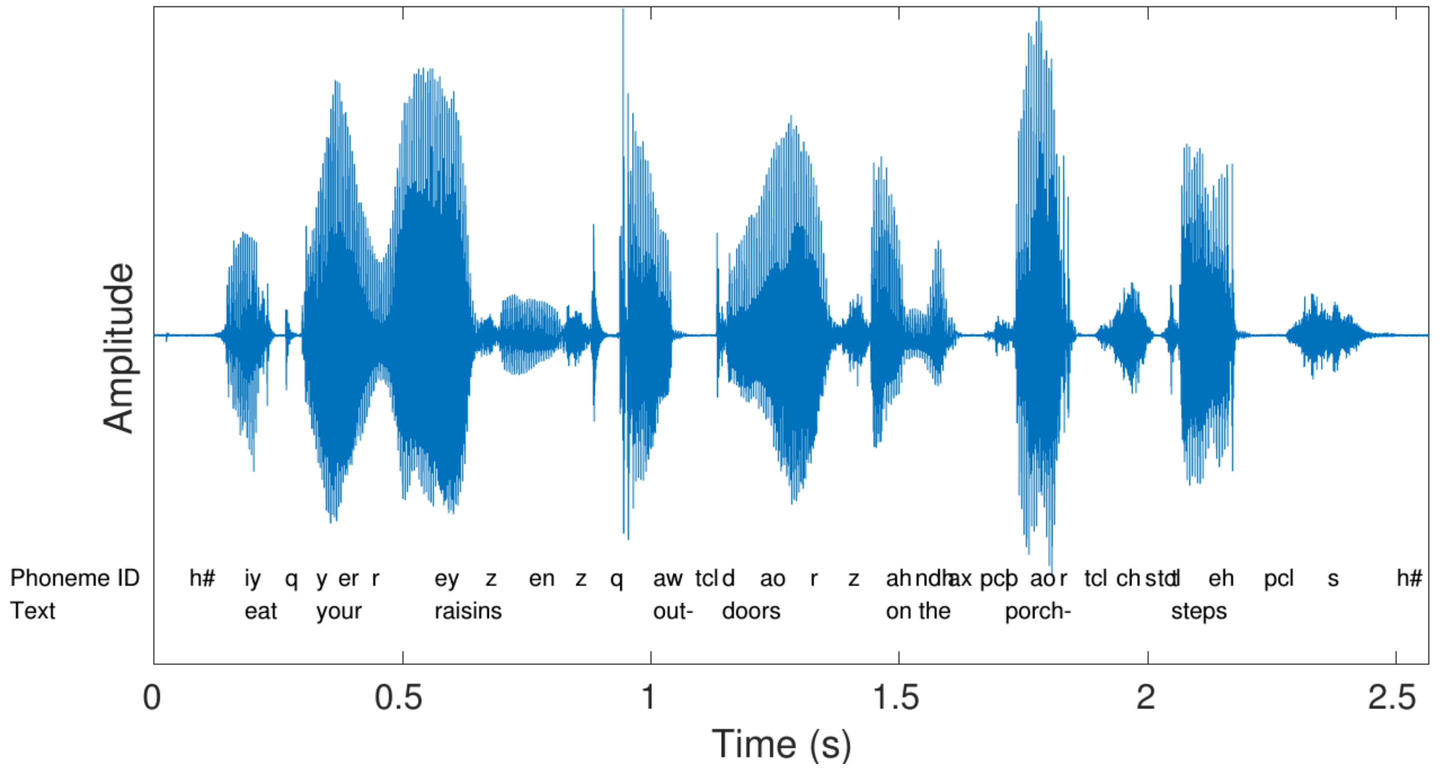
Sequence Models

What is a sequence?

- “This morning I took the dog for a walk”. Sentence

- 

Medical signals

- 

Speech waveform

A sequence modeling problem

predict the next word

A sequence modeling problem

“This morning I took the dog for a walk.”

A sequence modeling problem

"This morning I took the dog for a walk."

Given these words

Predict what comes next

An idea: use a fixed window

"This morning I took the dog for a walk."

Given these two words
predict the next word

An idea: use a fixed window

"This morning I took the dog for a walk."

Given these two words
predict the next word

[1 0 0 0 0 0 1 0 0 0]

for a

One hot feature vector
Indicates what
each word is

Prediction

Problem: we can’t model long term dependencies

“In Finland, I had a great time and I learnt some of the ________ language”

We need information from the far past and the future to accurately predict the correct word.

An idea: use entire sequence, as a set of counts

"This morning I took the dog for a walk.

"bag of words"

[0 1 0 0 1 0 0 … 0 0 1 1 0 0 0 1]

↓

Prediction

Problem…

Counts do not preserve the order.

Hence we lose all the sequential information! ☹

Problem…

Counts do not preserve the order.

Hence we lose all the sequential information! ☹

“The food was good, not bad at all”

“The food was bad, not good at all”

An idea: use a really big fixed window

"This morning I took the dog for a walk."

Given these 7 words
predict the next word

[1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 …]

morning I took the dog

Prediction

Problem: no parameter sharing

[0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 …]

this morning

each of these inputs has a separate parameter

[0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0…]

this morning

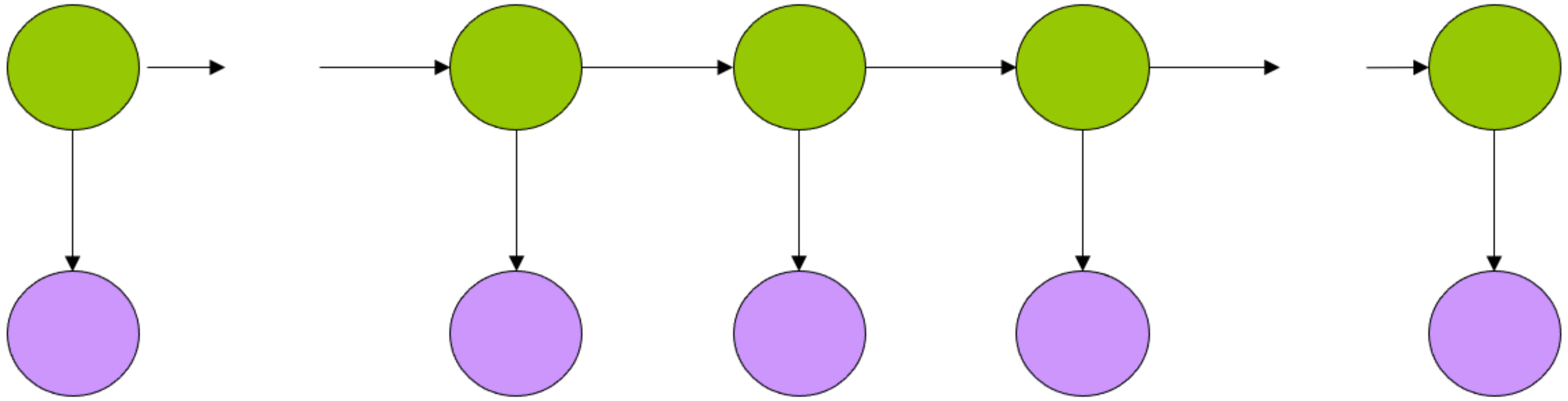
Things we learn about the sequence will not transfer if they appear at different points of the sequence.

To model sequences, we need…

- To deal with variable-length sequences
- To maintain sequence order
- To keep track of long term dependencies
- To share parameters across the sequence

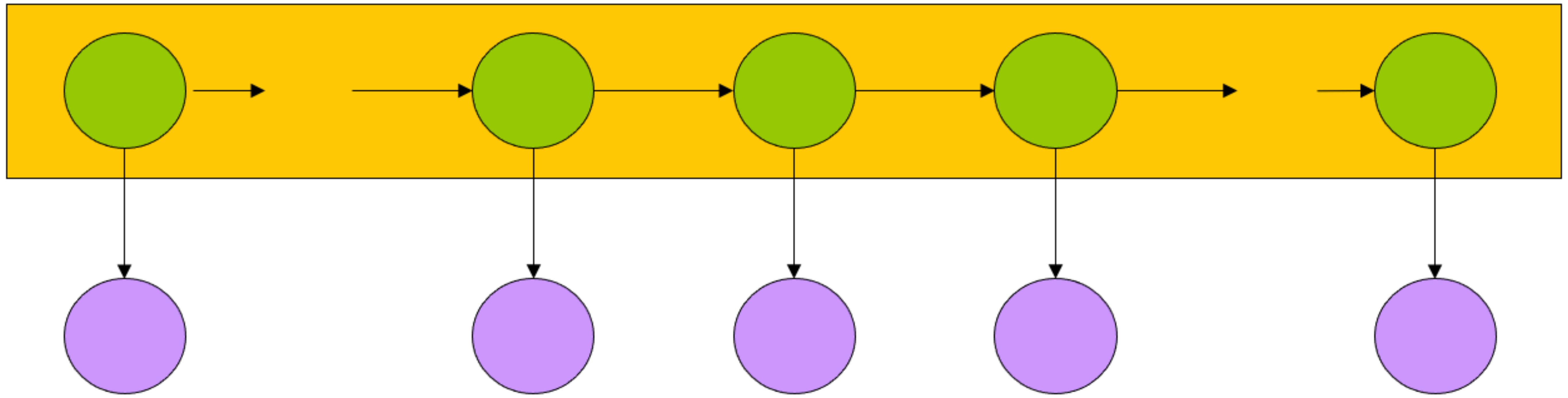
Hidden Markov Model (HMM)

What is an HMM?



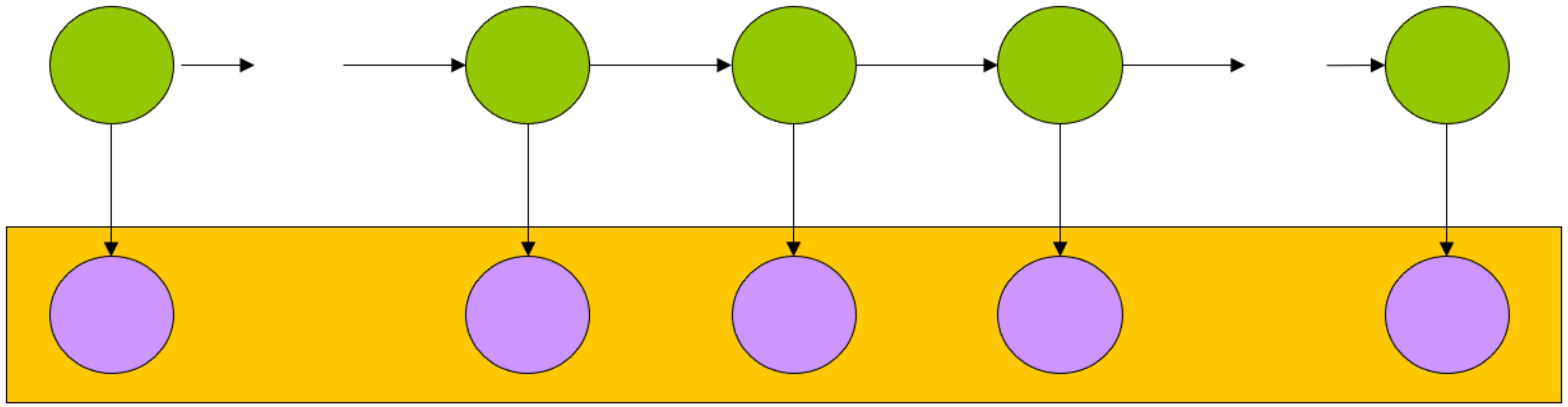
- Graphical Model
- Circles indicate states
- Arrows indicate probabilistic dependencies between states

What is an HMM?



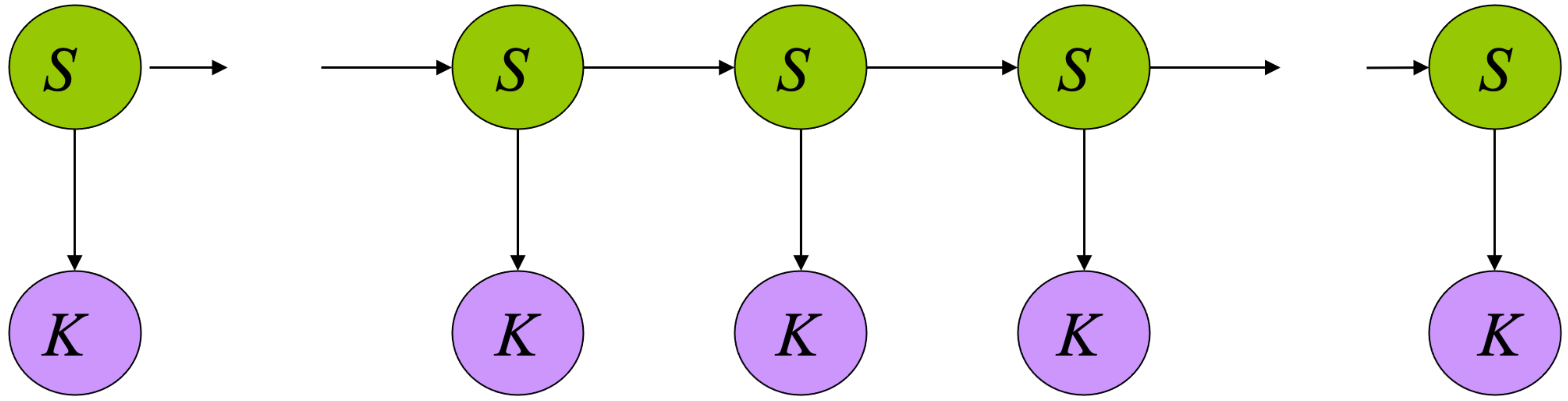
- Green circles are hidden states
- Dependent only on the previous state
- “The past is independent of the future given the present.”

What is an HMM?



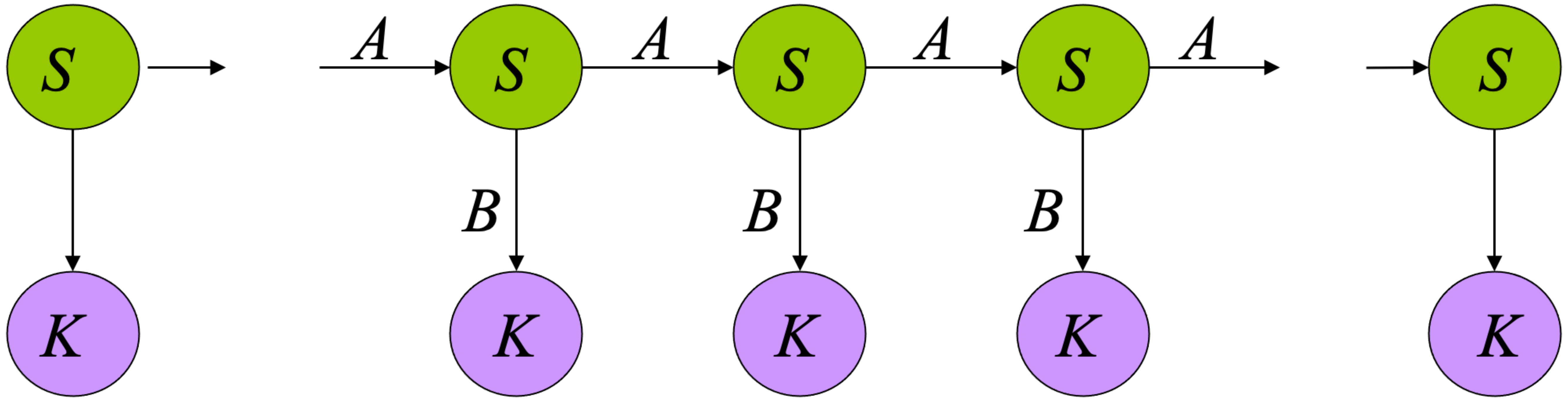
- Purple nodes are observed states
- Dependent only on their corresponding hidden state

HMM Notations



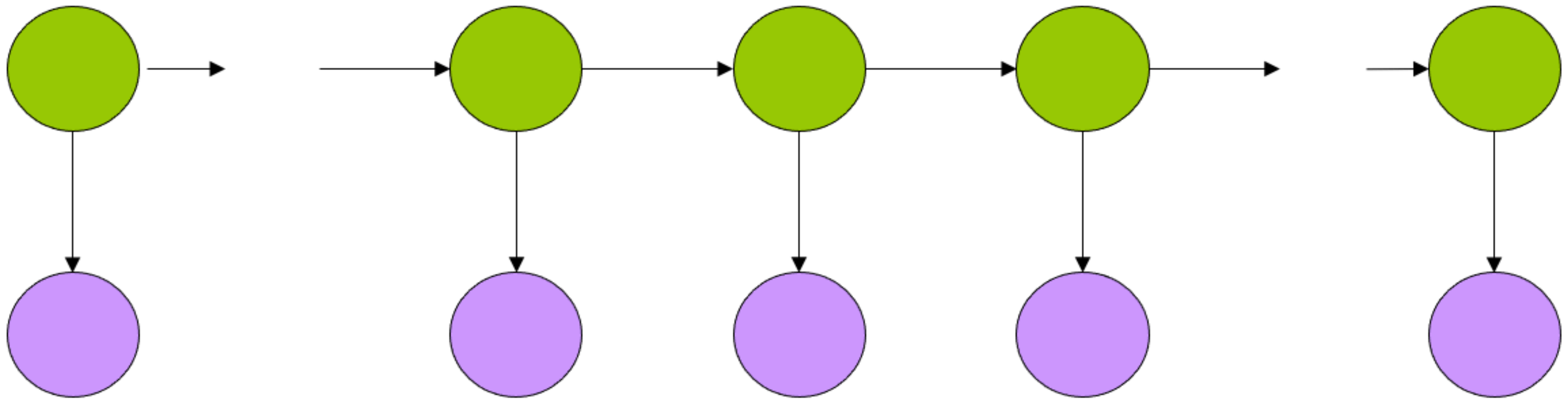
- $\{S, K, \Pi, A, B\}$
- S : $\{s_1 \ldots s_N\}$ are the values for the hidden states
- K : $\{k_1 \ldots k_M\}$ are the values for the observations

HMM Notations



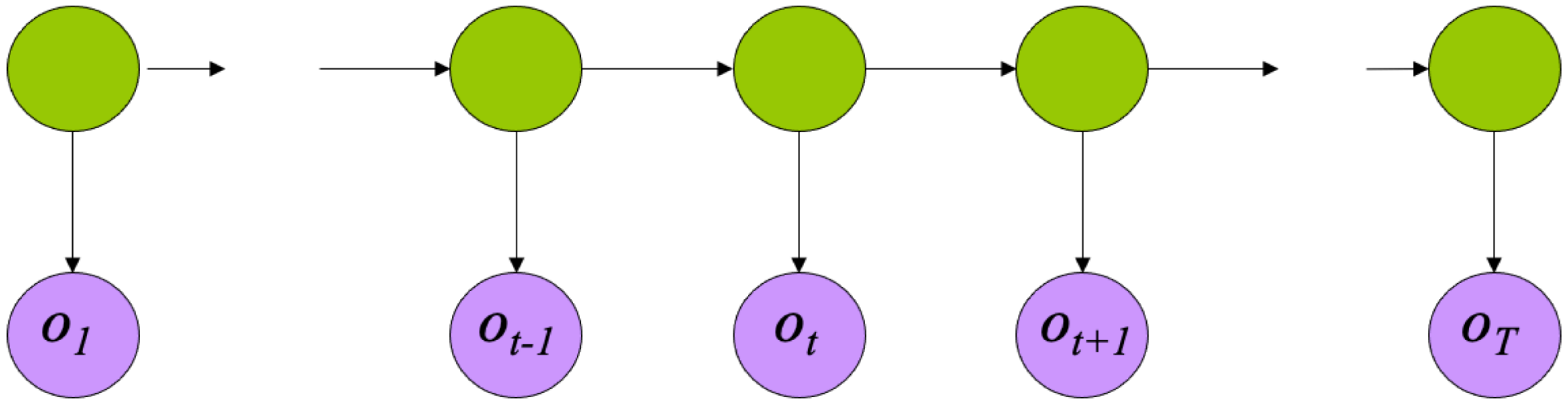
- $\{S, K, \Pi, A, B\}$
- $\Pi = \{\pi_\iota\}$ are the initial state probabilities
- $A = \{a_{ij}\}$ are the state transition probabilities
- $B = \{b_{ik}\}$ are the observation state probabilities

Inference in an HMM



- Compute the probability of a given observation sequence
- Given an observation sequence, compute the most likely hidden state sequence
- Given an observation sequence and set of possible models, which model most closely fits the data?

Decoding in HMM

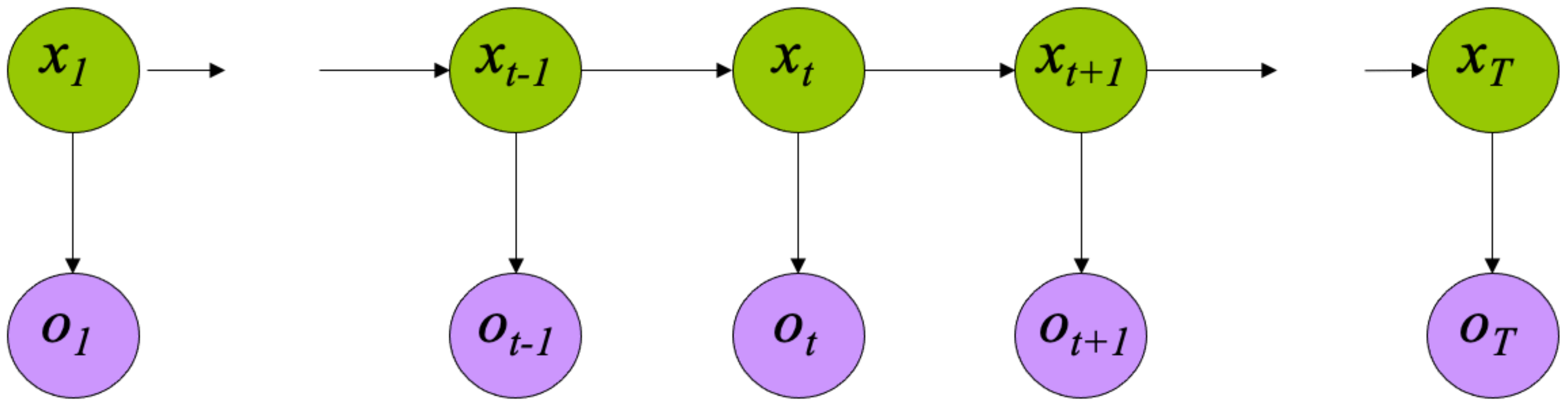


Given an observation sequence and a model,
compute the probability of the observation sequence

$$O = (o_1 \dots o_T),\ \mu = (A, B, \Pi)$$

Compute $P(O \mid \mu)$

Decoding in HMM



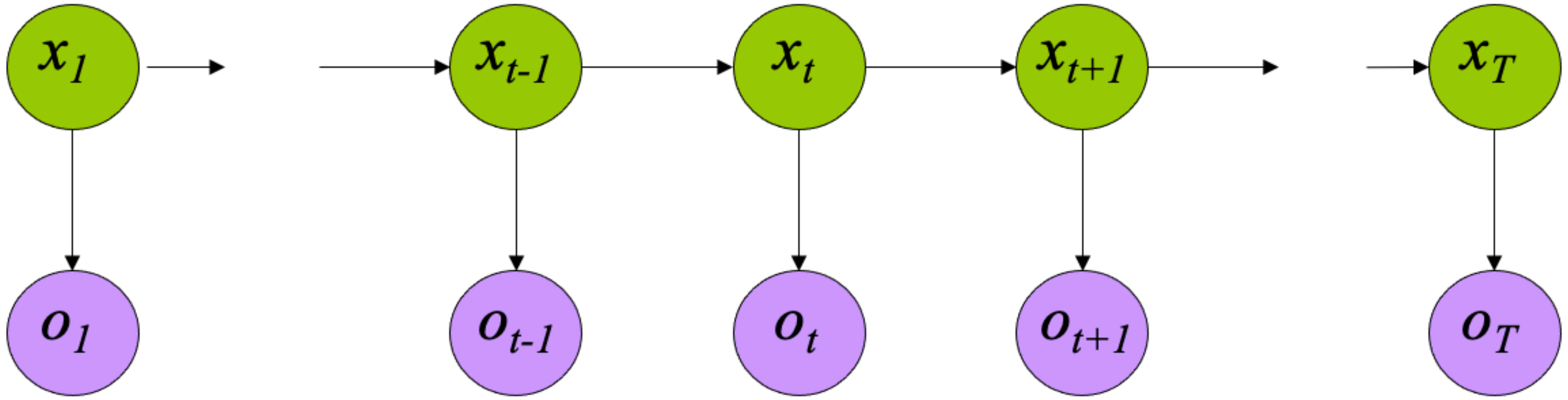
$$P(O \mid X, \mu) = b_{x_1 o_1} b_{x_2 o_2} \ldots b_{x_T o_T}$$

$$P(X \mid \mu) = \pi_{x_1} a_{x_1 x_2} a_{x_2 x_3} \ldots a_{x_{T-1} x_T}$$

$$P(O, X \mid \mu) = P(O \mid X, \mu) P(X \mid \mu)$$

$$\boxed{P(O \mid \mu) = \sum_X P(O \mid X, \mu) P(X \mid \mu)}$$

Decoding in HMM

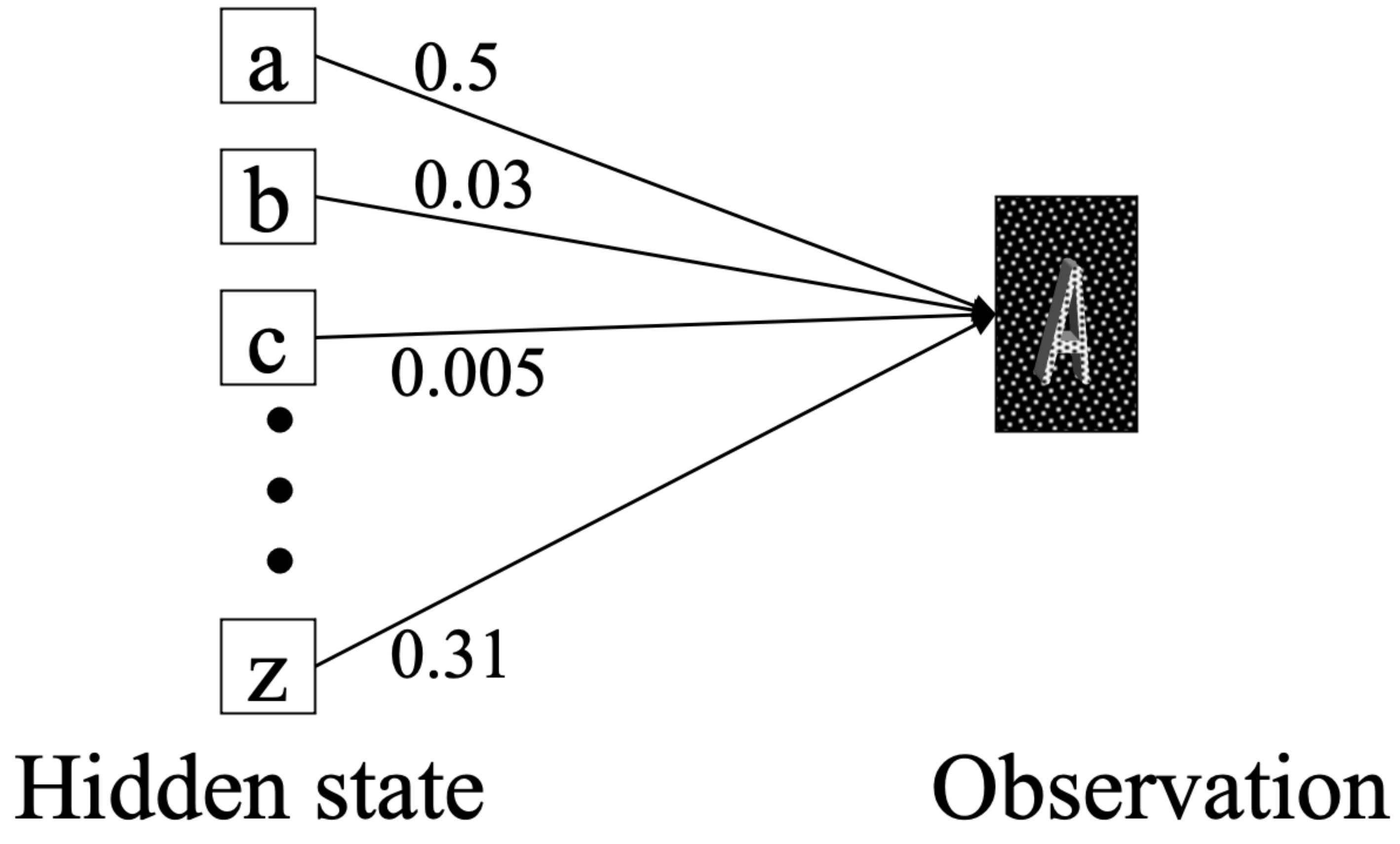


$$P(O \mid \mu) = \sum_{\{x_1 \ldots x_T\}} \pi_{x_1} b_{x_1 o_1} \prod_{t=1}^{T-1} a_{x_t x_{t+1}} b_{x_{t+1} o_{t+1}}$$

Example: Word recognition

- Typed word recognition, assume all characters are separated.

- Character recognizer outputs probability of the image being particular character, P(image|character).



Example: Word recognition

- Hidden states of HMM = characters.

- Observations = typed images of characters segmented from the images

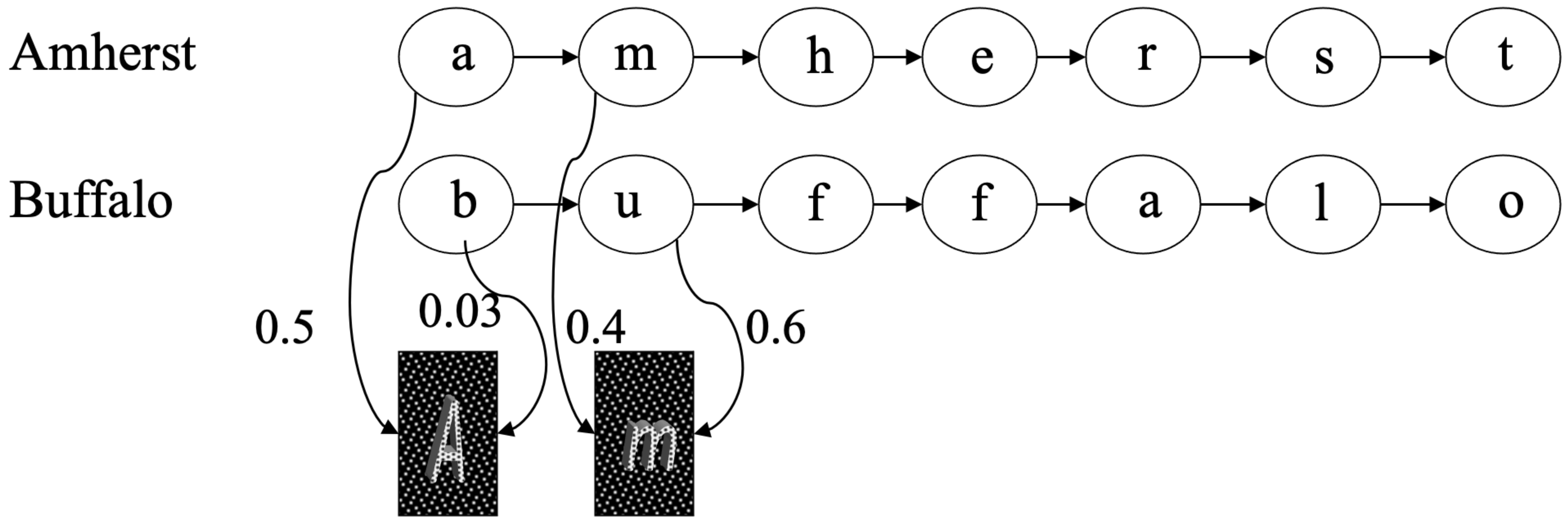
**Note that there is an infinite number of observations

- Observation probabilities = character recognizer scores.

- Transition probabilities will be defined differently in two subsequent models.

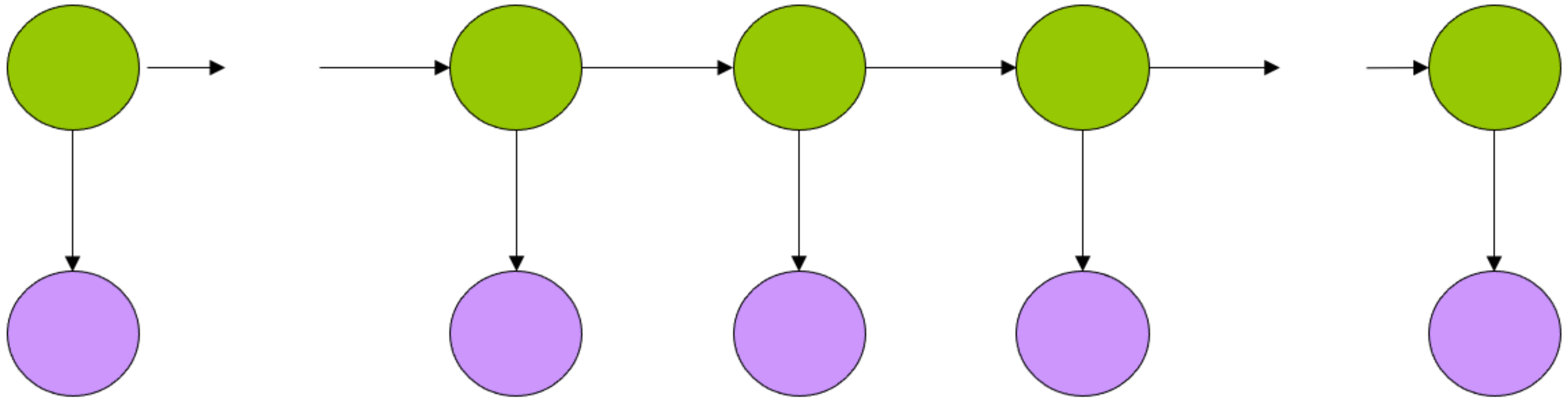
Example: Word recognition

- If lexicon is given, we can construct separate HMM models for each lexicon word.



- Here, recognition of word image is equivalent to the problem of evaluating few HMM models.

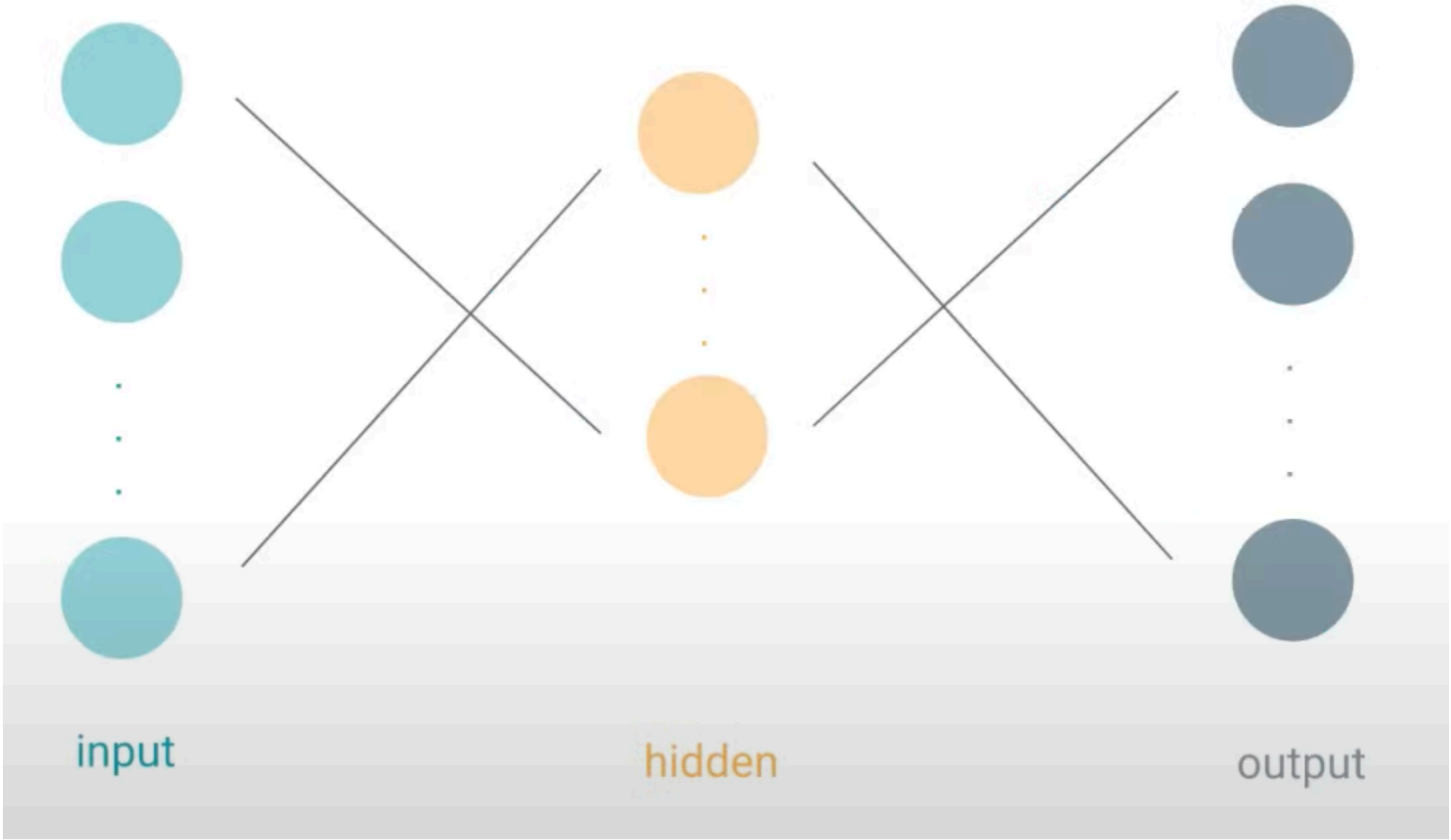
HMM Applications



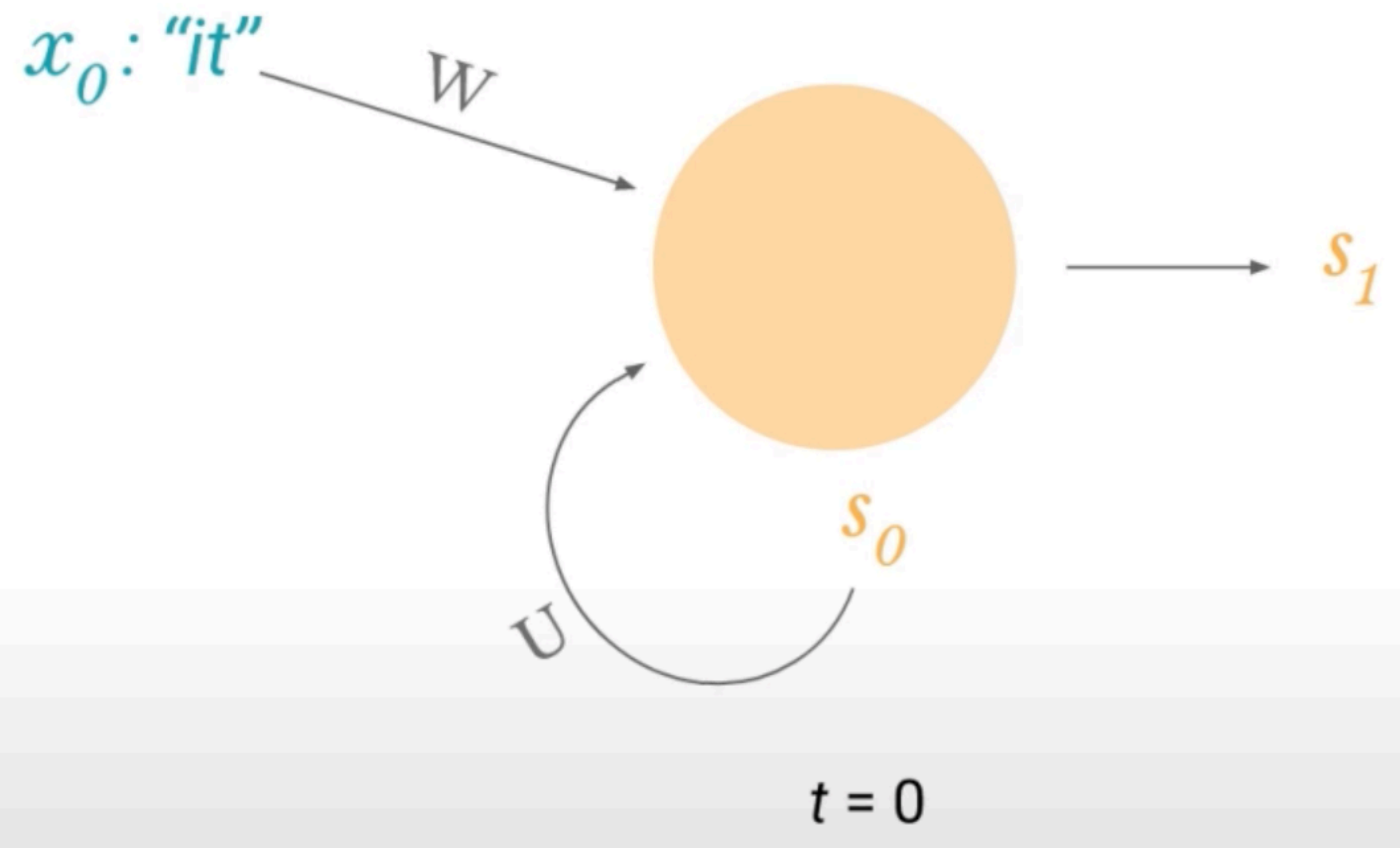
- Generating parameters for n-gram models
- Tagging speech
- Speech recognition

Recurrent Neural Networks (RNNs)

A neural network



RNNs remember their previous state

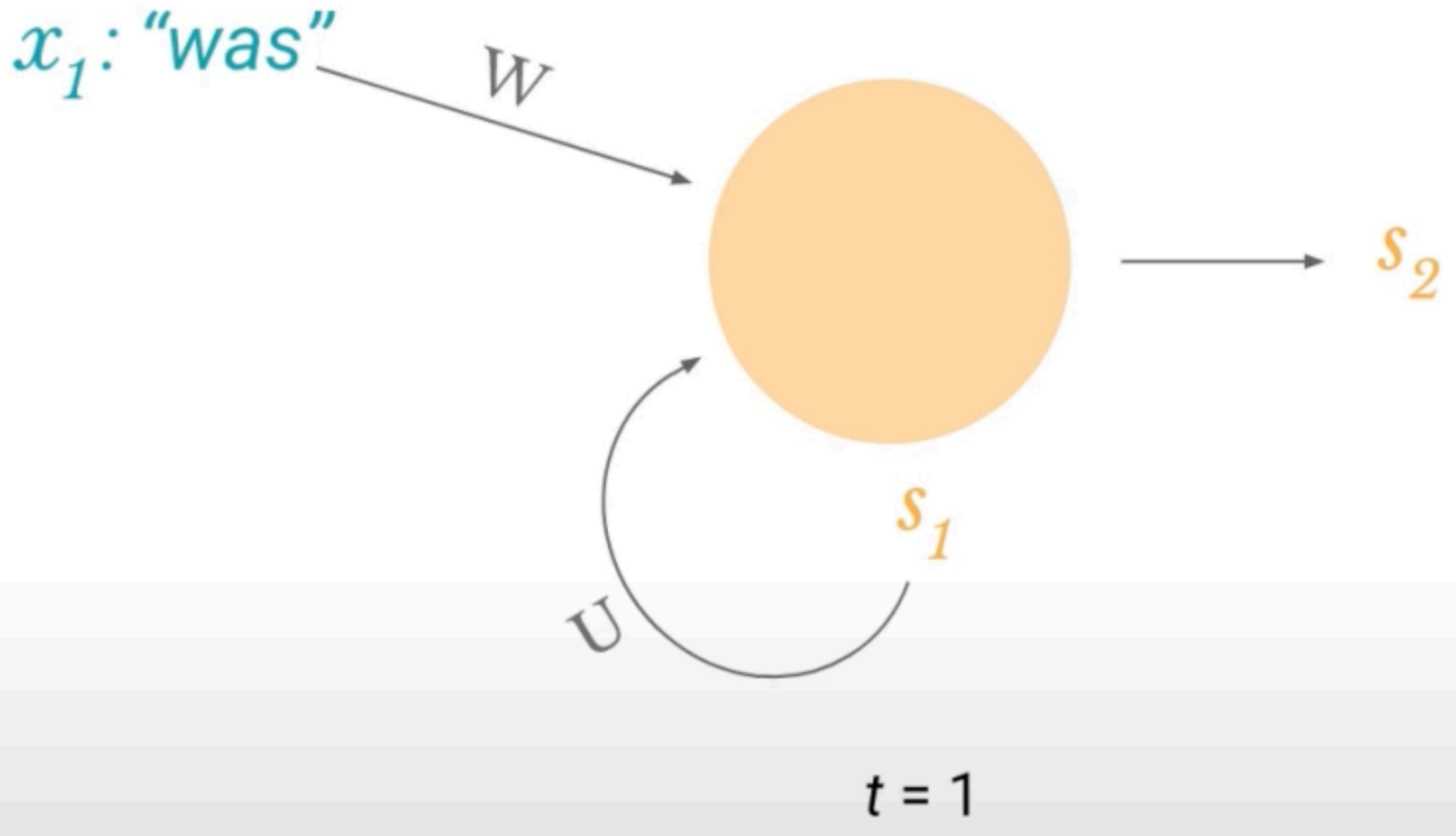


x_0 : vector representing first word
s_0 : cell state at $t = 0$ (some initialization)
s_1 : cell state at $t = 1$

$$s_1 = tanh(Wx_0 + Us_0)$$

$W,\ U$: weight matrices

RNNs remember their previous state

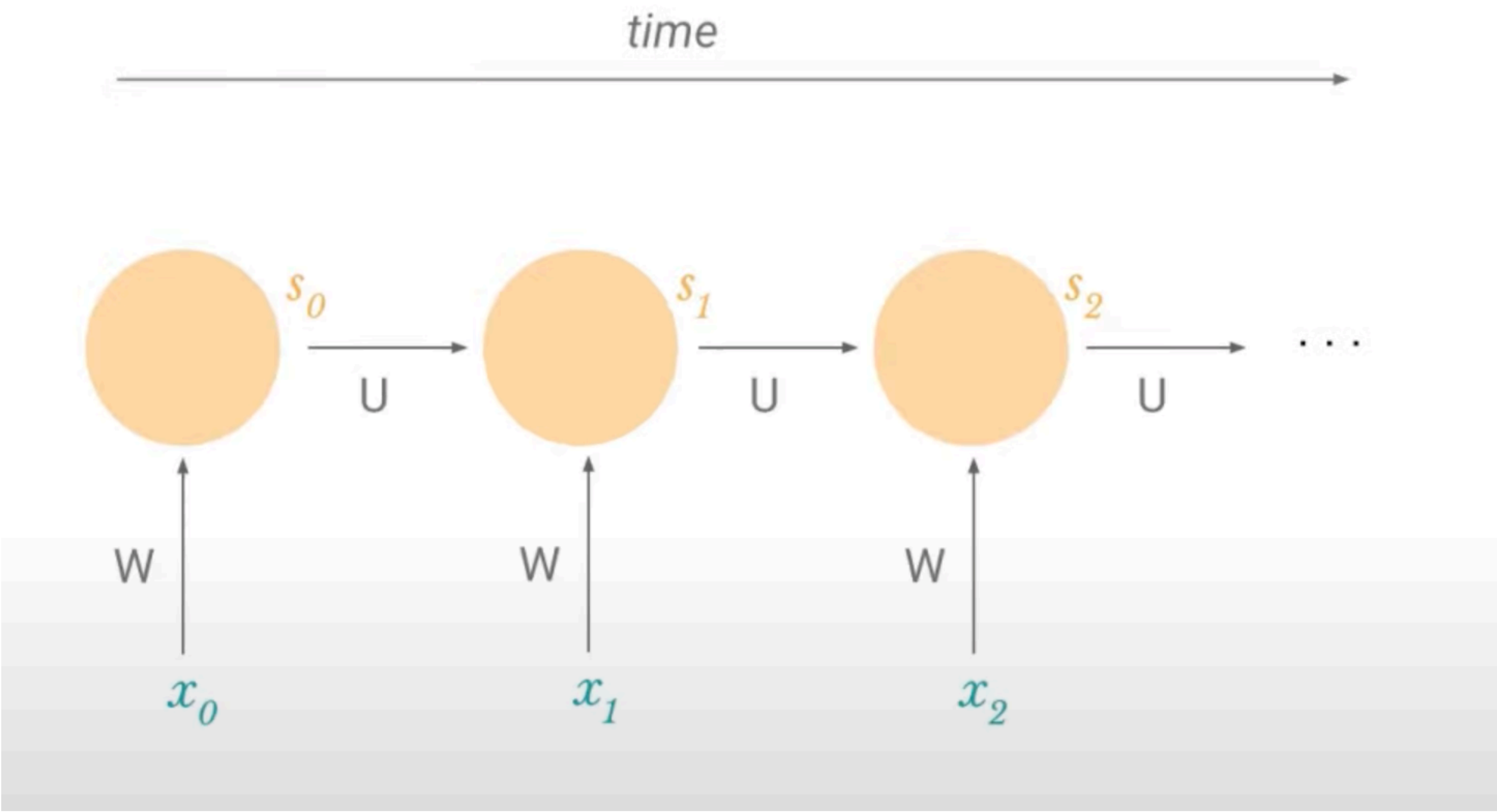


x_1 : vector representing second word
s_1 : cell state at $t = 1$
s_2 : cell state at $t = 2$

$$s_2 = tanh(Wx_1 + Us_1)$$

$W, \ U$: weight matrices

RNNs through time



To model sequences, we need…

- To deal with variable-length sequences
- To maintain sequence order
- To keep track of long term dependencies
- To share parameters across the sequence

Summary

- What is a sequence?
- Sequence modeling
- Hidden Markov Model (HMM)
- HMM Example
- A brief intro to RNNs

References

- Slides modified from “Sequence Modeling with Neural Networks” by Harini Suresh. 2018. MIT.
- Slides modified from “Hidden Markov Models” by David Meir Blei. 2009.