CS540 Introduction to Artificial Intelligence

Lecture 10

Young Wu

Based on lecture slides by Jerry Zhu, Yingyu Liang, and Charles Dyer

July 15, 2022

Secretary Problem

Quiz

- You are planning to interview 20 people in random order, after each interview, you have to either accept or reject the applicant before interviewing the next one. Applicants cannot be accepted after they are rejected. Suppose your strategy is to always reject the first n applicant and after that accept the first applicant who is better than all previous applicants. What is the n that maximizes the probability of accepting the best applicant among all 20?
  - $A : 0 - 4$
  - $B : 5 - 9$
  - $C : 10 - 14$
  - $D : 15 - 20$

$\rightarrow$ prob $= \frac{1}{e}$

$n = \frac{20}{e} \approx 7$

Q1

0.5 0.2 0.1 . 0.1 0.4 0.6

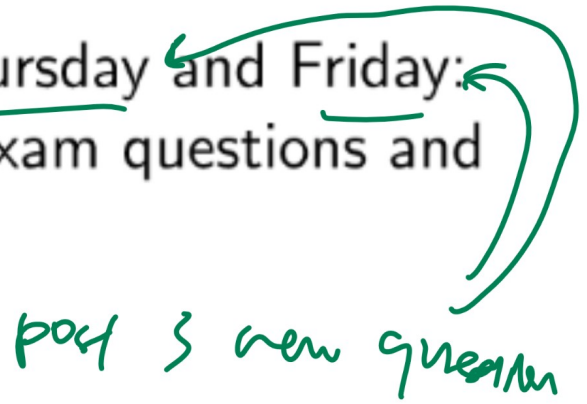
$n = 3$

R A

Guest Lectures and Review Sessions

Admin

- Asmit's guest lectures next Tuesday and Wednesday on reinforcement learning.
- Asmit's review sessions (optional) next Thursday and Friday: answer questions and going through past exam questions and the three new questions on the exam.



Exam Format 1

Admin

- Formula sheet on *W4* page (post on Piazza if you would like more).
- Open book, you are allowed to use the slides, Google search, etc, and you can write and use your own program to solve questions.
- You are not allowed to communicate with other students (questions randomly generated based on ID).

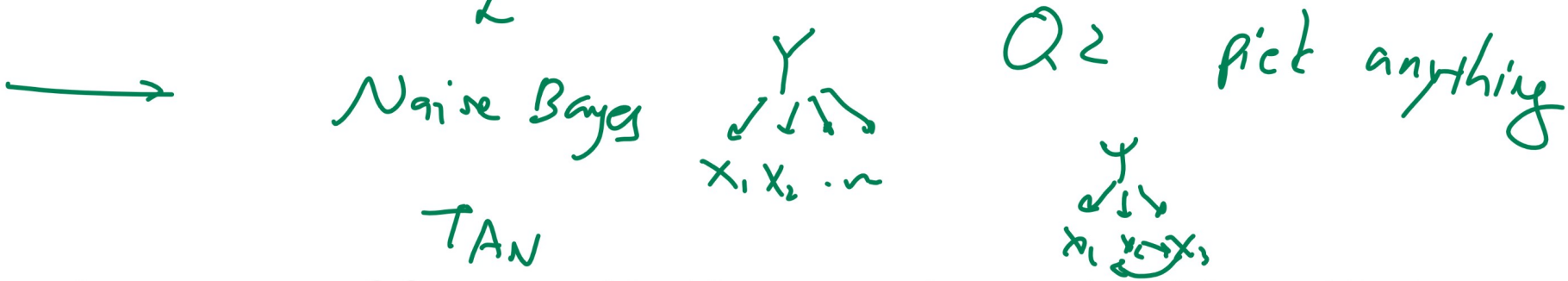
Exam Format 2

Admin

- The Canvas assignment XM1 will contain the link to the exam: available at 1 : 00 (message me if you would like to start at 12 : 45).
- You must join Zoom if you complete the exam online at home.
- No hints, no auto-grading for the exam questions, submit on the webpage AND on Canvas (make sure you leave enough time to submit before 2 : 15), if you submit late, you will get 0.

Special Bayesian Network for Sequences

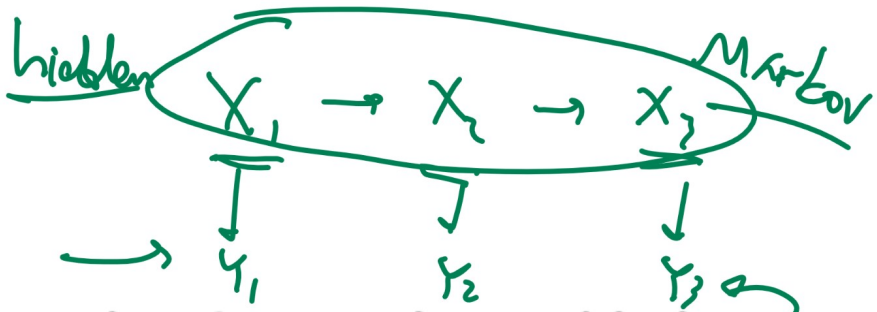
Motivation



- A sequence of features $X_1, X_2, \ldots$ can be modeled by a Markov Chain but they are not observable.
- A sequence of labels $Y_1, Y_2, \ldots$ depends only on the current hidden features and they are observable.
- This type of Bayesian Network is called a Hidden Markov Model.

HMM Applications Part 1

Motivation



- Weather prediction.
- Hidden states: $X_1, X_2, \ldots$ are weather that is not observable by a person staying at home (sunny, cloudy, rainy).
- Observable states: $Y_1, Y_2, \ldots$ are Badger Herald newspaper reports of the condition (dry, dryish, damp, soggy).
- Speech recognition.
- Hidden states: $X_1, X_2, \ldots$ are words.
- Observable states: $Y_1, Y_2, \ldots$ are acoustic features.

bigram model

$Pr[H, D | J, A]$

HMM Applications Part 2

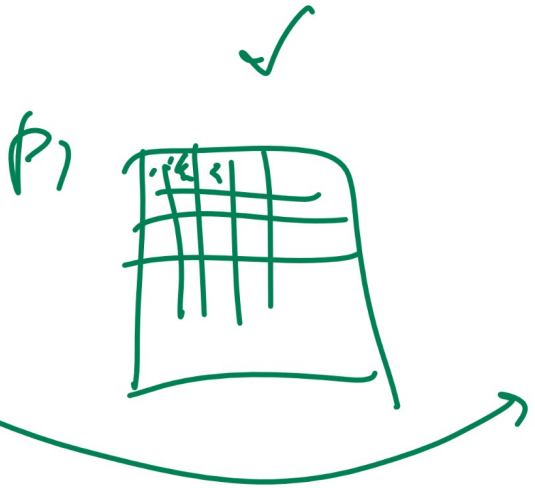
Motivation

- Stock or bond prediction.
- Hidden states: $X_1, X_2, \ldots$ are information about the company (profitability, risk measures).
- Observable states: $Y_1, Y_2, \ldots$ are stock or bond prices.
- Speech synthesis: Chatbox.
- Hidden states: $X_1, X_2, \ldots$ are context or part of speech.
- Observable states: $Y_1, Y_2, \ldots$ are words.

Other HMM Applications

Motivation

- Machine translation.
- Handwriting recognition.
- Gene prediction.
- Traffic control.



Hidden Markov Model Diagram

Motivation

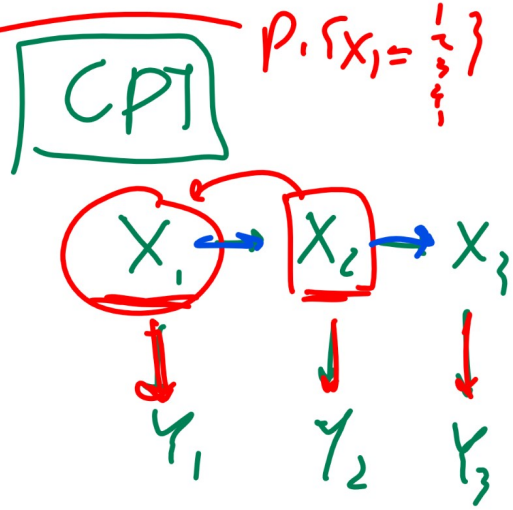
Transition and Likelihood Matrices

Motivation

- An initial distribution vector and two-state transition matrices are used to represent a hidden Markov model.

1. Initial state vector: π.

$$\pi_i = \mathbb{P}\{X_1 = i\}, i \in 1, 2, ..., |X|$$



2. State transition matrix: A.

$$A_{ij} = \mathbb{P}\{X_t = j | X_{t-1} = i\}, i, j \in 1, 2, ..., |X|$$

$P\{X_t | X_{t-1}\}$

3. Observation Likelihood matrix (or output probability distribution): B.

$P\{Y_t | X_t\}$

$$B_{ij} = \mathbb{P}\{Y_t = i | X_t = j\}, i \in 1, 2, ..., |Y|, j \in 1, 2, ..., |X|$$

Evaluation and Training

Motivation

- There are three main tasks associated with an HMM.

1. Evaluation problem: finding the probability of an observed sequence given an HMM: $y_1, y_2, ...$
2. Decoding problem: finding the most probable hidden sequence given the observed sequence: $x_1, x_2, ...$
3. Learning problem: finding the most probable HMM given an observed sequence: $\pi, A, B, ...$

CPT

Expectation-Maximization Algorithm

Description

- Start with a random guess of π, A, B.
- Compute the forward probabilities: the joint probability of an observed sequence and its hidden state.
- Compute the backward probabilities: the probability of an observed sequence given its hidden state.
- Update the model π, A, B using Bayes rule.
- Repeat until convergence.
- Sometimes, it is called the Baum-Welch Algorithm.

Hidden Markov Model Example 1

Definition

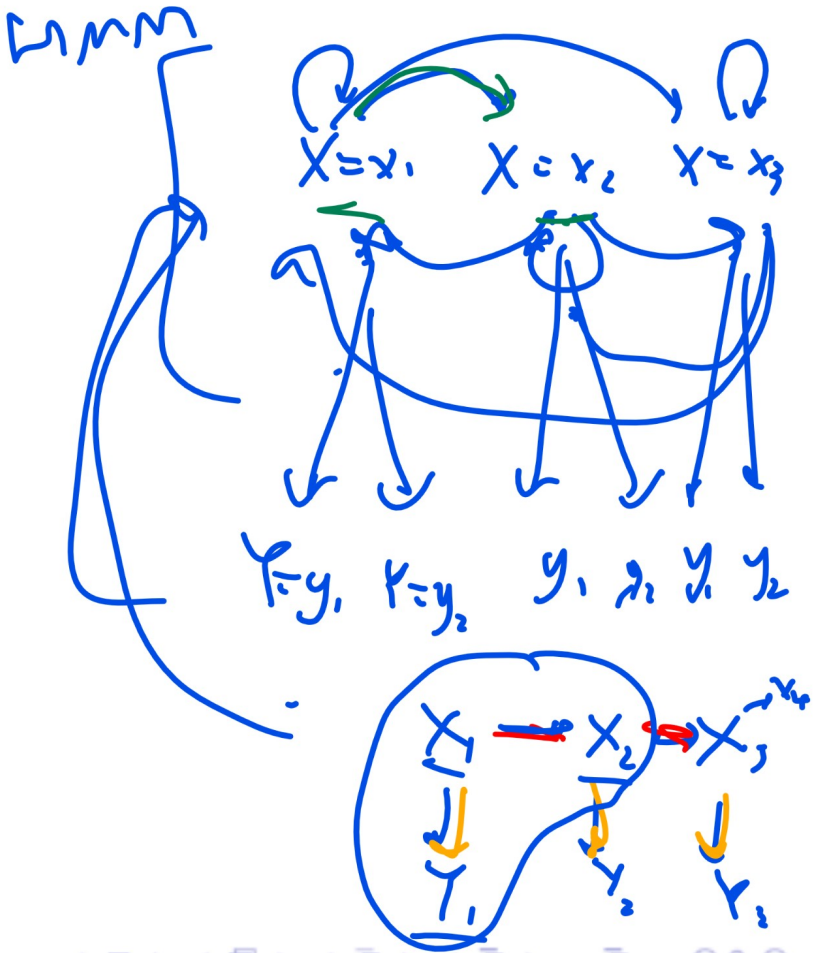
- Compute $\mathbb{P}\{X_4 = 1, X_5 = 2 | X_3 = 0\}$.

$$\Pr\{X_4 = 1 \mid X_3 = 0\}$$

$$\Pr\{X_5 = 2 \mid X_4 = 1, X_3 = 0\}$$

$$P_{0\to 1} \cdot P_{1\to 2}$$

0.2 0.3



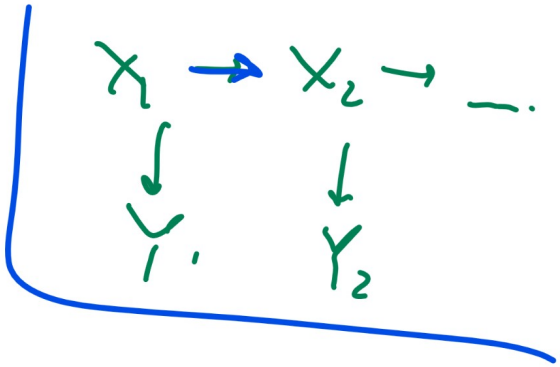
Hidden Markov Model Example 1 Computations

Definition

Hidden Markov Model Example 2

Definition

- Compute $\mathbb{P}\{Y_1 = 0, Y_2 = 1\}$.



$= \Pr\{Y_1 = 0, Y_2 = 1, X_1 = \begin{matrix}0\\1\\2\end{matrix}, X_2 = \begin{matrix}0\\1\\2\end{matrix}\}$

↓ each term

9 prob

$\Pr\{Y_1 = 0 \mid X_1 = 0\} \cdot \Pr\{Y_2 = 1 \mid X_2 = 0\} \cdot \Pr\{X_1 = 0\}$

0.47 0.53 0.38

$\Pr\{X_2 = 0 \mid X_1 = 0\}$

Hidden Markov Model Example 2 Computations

Definition

Hidden Markov Model Example 3

Definition

- Compute $\mathbb{P}\{X_1 = 0, X_2 = 2 | Y_1 = 0, Y_2 = 1\}$.

CPT

$P_r\{X_1=0\} \cdot P_r\{X_2=2 | X_1=0\}$

$P\{Y_1=1 | X_1=0\}$

$P_r\{Y_2=1 | X_2=2\}$

$$\frac{P_r\{X_1=0, X_2=2, Y_1=0, Y_2=1\}}{P_r\{Y_1=0, Y_2=1\}}$$

previous Q.

Hidden Markov Model Example 3 Computations

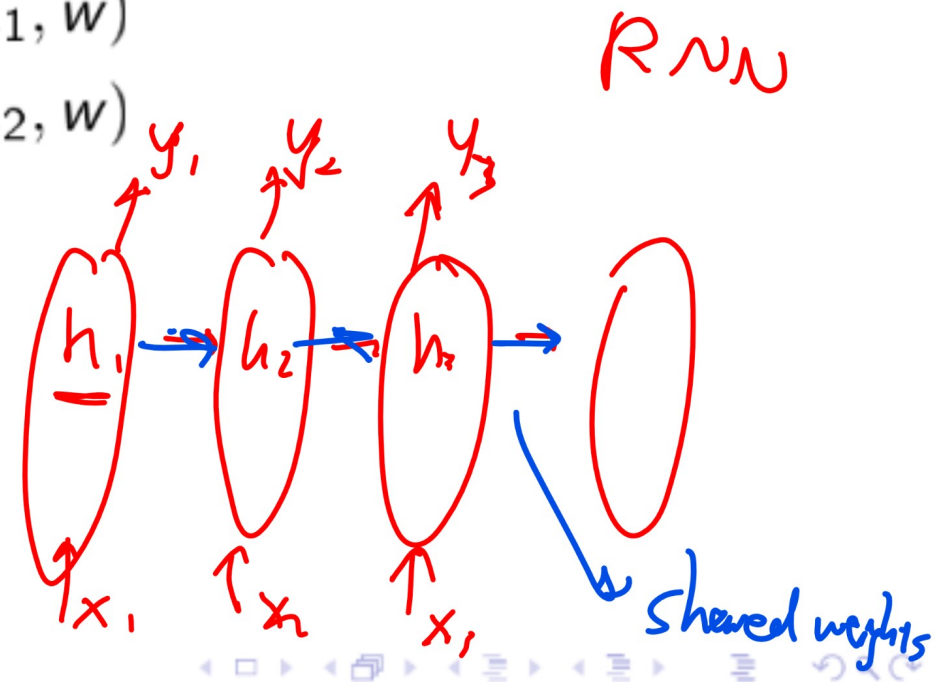
Definition

Dynamic System

Motivation

- The hidden units are used as the hidden states.
- They are related by the same function over time.

$$h_{t+1} = f(h_t, w)$$
$$h_{t+2} = f(h_{t+1}, w)$$
$$h_{t+3} = f(h_{t+2}, w)$$
$$\dots$$

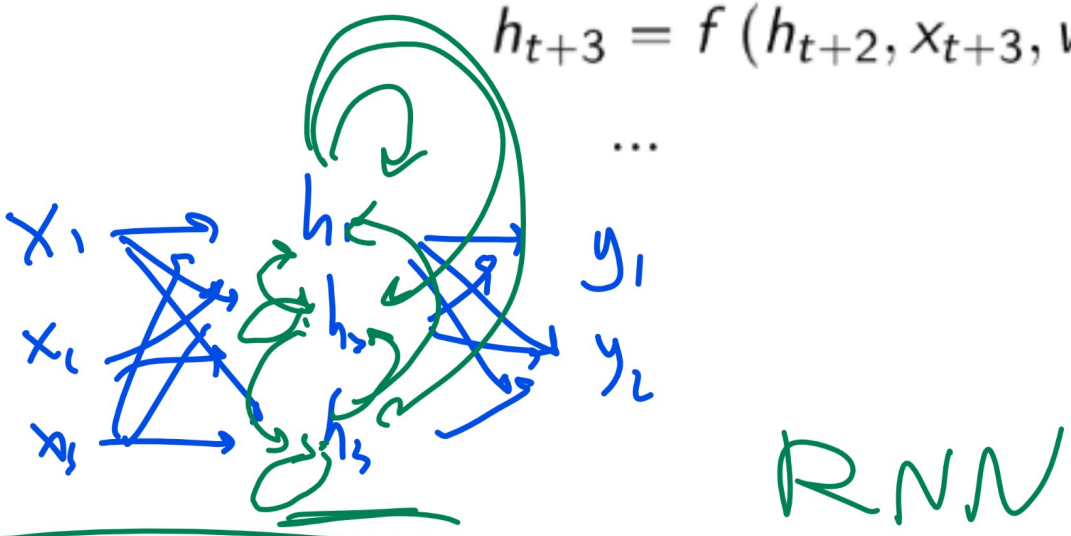


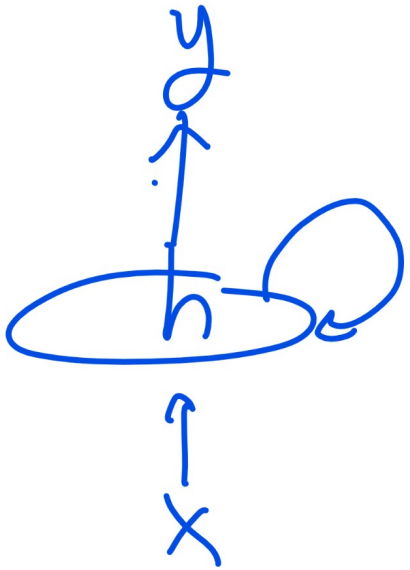
Dynamic System with Input

Motivation

- The input units can also drive the dynamics of the system.
- They are still related by the same function over time.

$$h_{t+1} = f(h_t, x_{t+1}, w)$$
$$h_{t+2} = f(h_{t+1}, x_{t+2}, w)$$
$$h_{t+3} = f(h_{t+2}, x_{t+3}, w)$$
$$\dots$$





Dynamic System with Output

Motivation

- The output units only depend on the hidden states.

$$y_{t+1} = f(h_{t+1})$$
$$y_{t+2} = f(h_{t+2})$$
$$y_{t+3} = f(h_{t+3})$$
$$\dots$$

Dynamic System Diagram

Motivation

Recurrent Neural Network Structure Diagram

Motivation

Activation Functions

Definition

- The hidden layer activation function can be the tanh activation, and the output layer activation function can be the softmax function.

$$z_t^{(x)} = W^{(x)} x_t + W^{(h)} a_{t-1}^{(x)} + b^{(x)}$$

$$a_t^{(x)} = g\left(z_t^{(x)}\right), g\left(\boxed{\cdot}\right) = \tanh\left(\boxed{\cdot}\right)$$

$$z_t^{(y)} = W^{(y)} a_t^{(x)} + b^{(y)}$$

$$a_t^{(y)} = g\left(z_t^{(y)}\right), g\left(\boxed{\cdot}\right) = \text{softmax}\left(\boxed{\cdot}\right)$$

Cost Functions

Definition

- Cross entropy loss is used with softmax activation as usual.

$$C_t = H\left(y_t, a_t^{(y)}\right)$$

$$C = \sum_t C_t$$

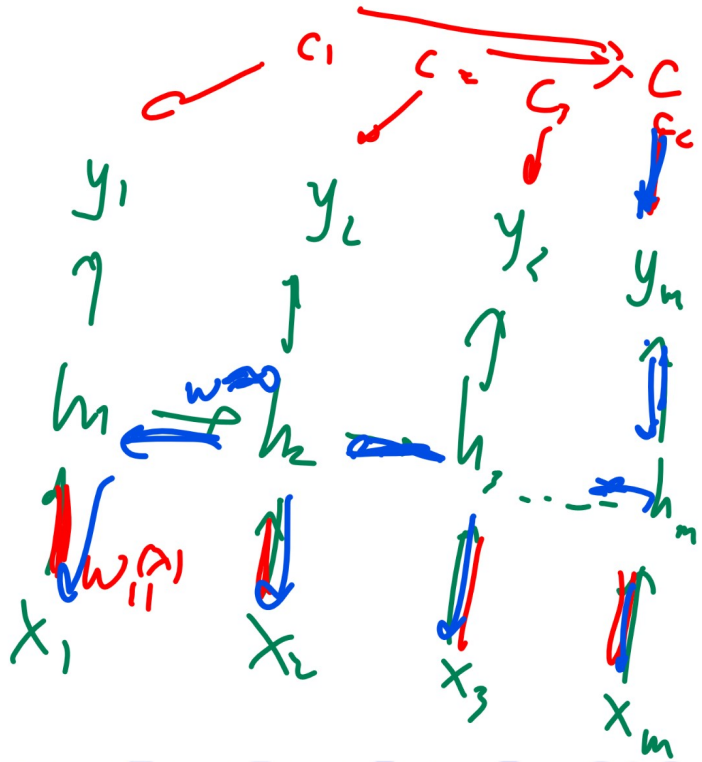
BackPropogation Through Time

Definition

- The gradient descent algorithm for recurrent neural networks is called BackPropogation Through Time (BPTT). The update procedure is the same as standard neural networks using the chain rule.

$$w = w - \alpha \frac{\partial C}{\partial w}$$

$$b = b - \alpha \frac{\partial C}{\partial b}$$



Unfolded Network Diagram

Definition

Vanishing and Exploding Gradient

Discussion

- If the weights are small, the gradient through many layers will shrink exponentially. This is called the vanishing gradient problem.
- If the weights are large, the gradient through many layers will grow exponentially. This is called the exploding gradient problem.
- Fully connected and convolutional neural networks only have a few hidden layers, so vanishing and exploding gradient is not a problem in training those networks.
- In a recurrent neural network, if the sequences are long, the gradients can easily vanish or explode.

RNN Variants

Discussion

- Long Short Term Memory (LSTM): gated units to keep track of long term dependencies.
- Gated Recurrent Unit (GRU): different gated units.
- Transformers (BERT, GPT): no recurrent units, positional encoding, attention mechansim.

RNN

D3.

D3 Too

DALL·E

x_1

word position in sentence.