



**Natalie Parde, Ph.D.**

Department of Computer  
Science

University of Illinois at  
Chicago

CS 521: Statistical Natural  
Language Processing

Spring 2020

**Exam Review**

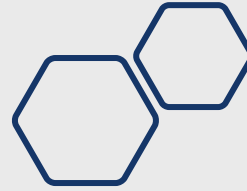
# Exam Format

- True/False (30%)
  - 15 questions, 2 points each
  - No partial credit (autograded)
- Multiple Choice (40%)
  - 8 questions, 5 points each
  - No partial credit (autograded)
- Problem Solving (30%)
  - 3 questions, 10 points each
  - Show work for partial credit (graded manually)
- Bonus Question (10%)
  - Problem solving-ish question
  - Points added to exam score (max exam score = 100)

# Sample Exam

- Currently available on Piazza
- Solution will be posted after class

# What should I study?



- Questions designed based on **slides**
- Problem solving questions similar (not identical!) to **problem solving examples** in slides



# What content will the exam cover?

---

language modeling

---

data collection

---

logistic regression

---

word vectors

---

feedforward neural networks

---

backpropagation

---

convolutional neural networks

---

recurrent neural networks

---

LSTMs

---

GRUs

---

encoder-decoder models

---

attention

---

coreference resolution

---

discourse coherence

---

**What will I  
for sure *not*  
need to  
memorize?**

- Derivatives for specific activation functions
- Log values

# What should I bring to the exam?

- 
- Pen or pencil
  - UIN (you'll need to write it on the first page of the exam)
  - This exam will be:
    - Closed note
    - Closed book
    - Closed device
  - You will not need a calculator



# How long will the exam last?

Full class period, if needed (75 minutes)







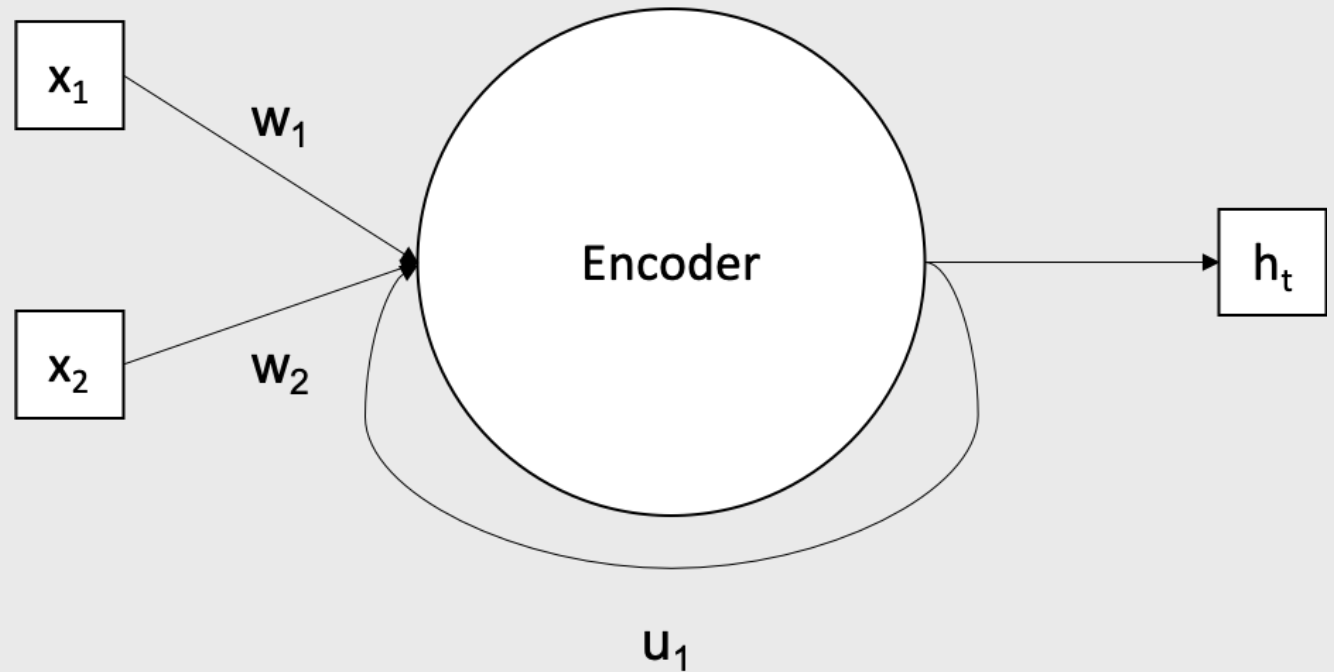
**Solution  
Time!**

**True/False**

**Multiple Choice**

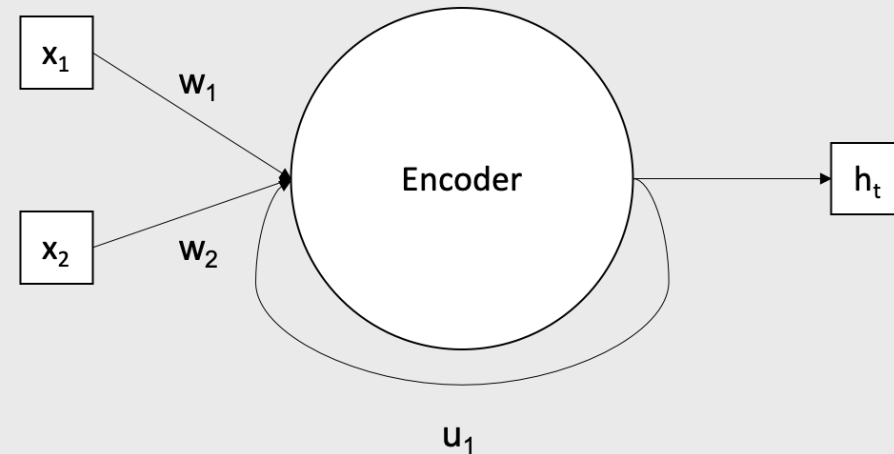
# Problem-Solving Questions

Assume we have an extremely simple encoder-decoder model, designed to translate a sarcastic sentence into its non-sarcastic counterpart. The encoder (pictured below) is comprised of a single RNN unit. Input tokens are represented as two-dimensional word embeddings, for which the encoder has learned weights  $w = [1.0, 2.0]$ . The encoder has also learned to weight the output of the RNN at the previous timestep as  $u = [5.0]$ . The unit uses a ReLU activation function.



# Problem-Solving Questions

Assume we have an extremely simple encoder-decoder model, designed to translate a sarcastic sentence into its non-sarcastic counterpart. The encoder (pictured below) is comprised of a single RNN unit. Input tokens are represented as two-dimensional word embeddings, for which the encoder has learned weights  $w = [1.0, 2.0]$ . The encoder has also learned to weight the output of the RNN at the previous timestep as  $u = [5.0]$ . The unit uses a ReLU activation function.

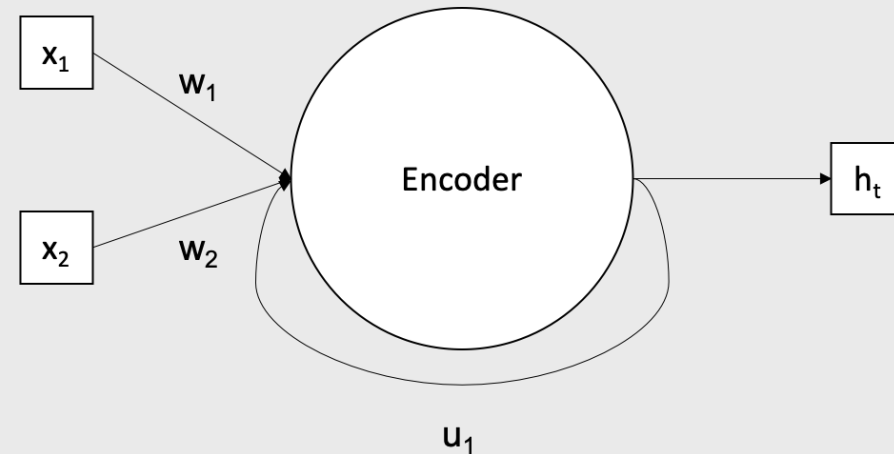


The following word embedding dictionary is available:

word	$x_1$	$x_2$
i	2	0
loooooove	1	2
midterms	2	1

# Problem-Solving Questions

Assume we have an extremely simple encoder-decoder model, designed to translate a sarcastic sentence into its non-sarcastic counterpart. The encoder (pictured below) is comprised of a single RNN unit. Input tokens are represented as two-dimensional word embeddings, for which the encoder has learned weights  $w = [1.0, 2.0]$ . The encoder has also learned to weight the output of the RNN at the previous timestep as  $u = [5.0]$ . The unit uses a ReLU activation function.



The following word embedding dictionary is available:

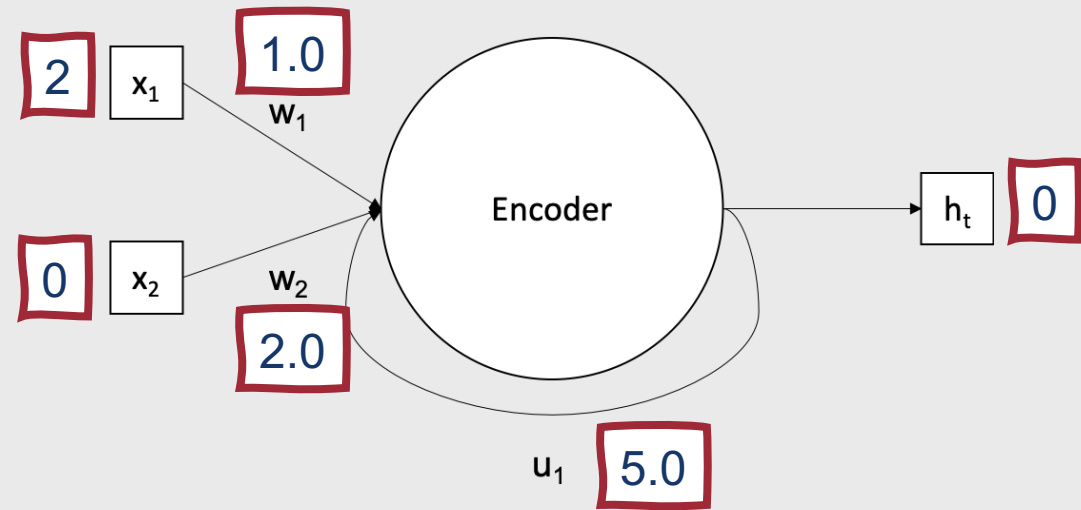
word	$x_1$	$x_2$
i	2	0
loooooove	1	2
midterms	2	1

Encode the sequence below:

*i loooooove midterms*

# Problem-Solving Questions

Assume we have an extremely simple encoder-decoder model, designed to translate a sarcastic sentence into its non-sarcastic counterpart. The encoder (pictured below) is comprised of a single RNN unit. Input tokens are represented as two-dimensional word embeddings, for which the encoder has learned weights  $w = [1.0, 2.0]$ . The encoder has also learned to weight the output of the RNN at the previous timestep as  $u = [5.0]$ . The unit uses a ReLU activation function.



The following word embedding dictionary is available:

word	$x_1$	$x_2$
i	2	0
loooooove	1	2
midterms	2	1

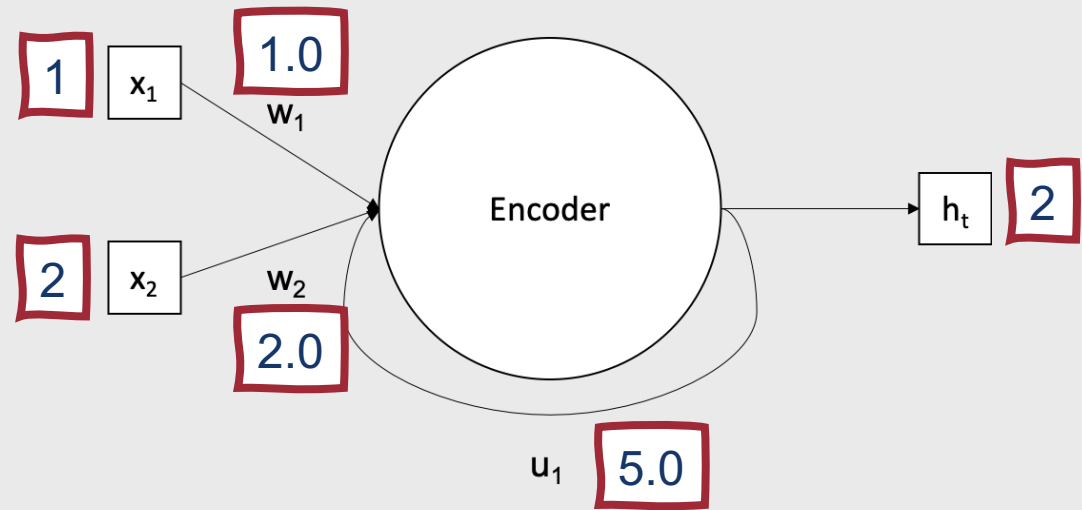
Encode the sequence below:

*i loooooove midterms*

$$h_1 = \text{ReLU}((2 * 1.0 + 0 * 2.0) + 0 * 5.0) = \text{ReLU}(2) = 2$$

# Problem-Solving Questions

Assume we have an extremely simple encoder-decoder model, designed to translate a sarcastic sentence into its non-sarcastic counterpart. The encoder (pictured below) is comprised of a single RNN unit. Input tokens are represented as two-dimensional word embeddings, for which the encoder has learned weights  $w = [1.0, 2.0]$ . The encoder has also learned to weight the output of the RNN at the previous timestep as  $u = [5.0]$ . The unit uses a ReLU activation function.



The following word embedding dictionary is available:

word	$x_1$	$x_2$
i	2	0
loooooove	1	2
midterms	2	1

Encode the sequence below:

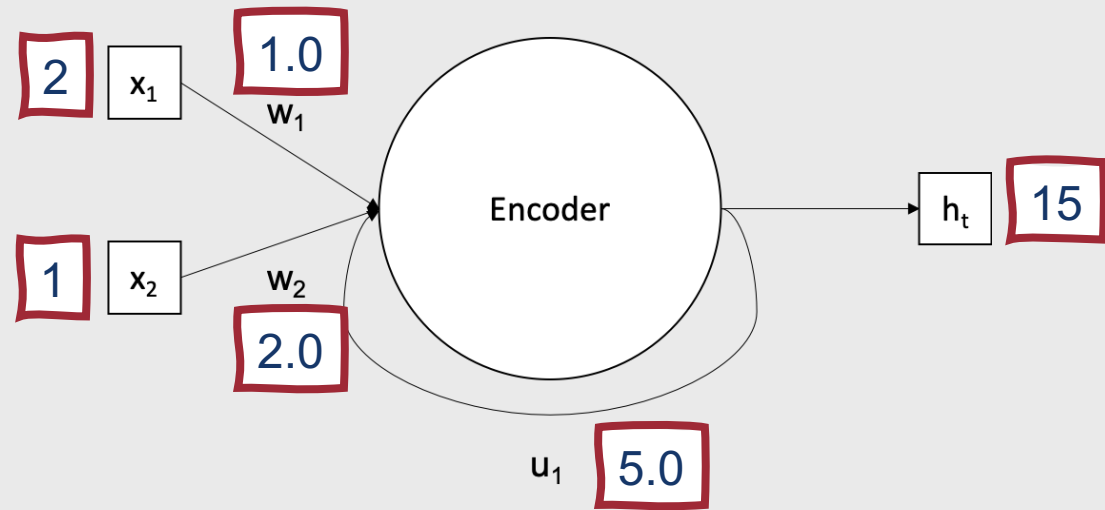
*i loooooove midterms*

$$h_1 = \text{ReLU}((2 * 1.0 + 0 * 2.0) + 0 * 5.0) = \text{ReLU}(2) = 2$$

$$h_2 = \text{ReLU}((1 * 1.0 + 2 * 2.0) + 2 * 5.0) = \text{ReLU}(15) = 15$$

# Problem-Solving Questions

Assume we have an extremely simple encoder-decoder model, designed to translate a sarcastic sentence into its non-sarcastic counterpart. The encoder (pictured below) is comprised of a single RNN unit. Input tokens are represented as two-dimensional word embeddings, for which the encoder has learned weights  $w = [1.0, 2.0]$ . The encoder has also learned to weight the output of the RNN at the previous timestep as  $u = [5.0]$ . The unit uses a ReLU activation function.



The following word embedding dictionary is available:

word	$x_1$	$x_2$
i	2	0
loooooove	1	2
midterms	2	1

Encode the sequence below:

*i loooooove midterms*

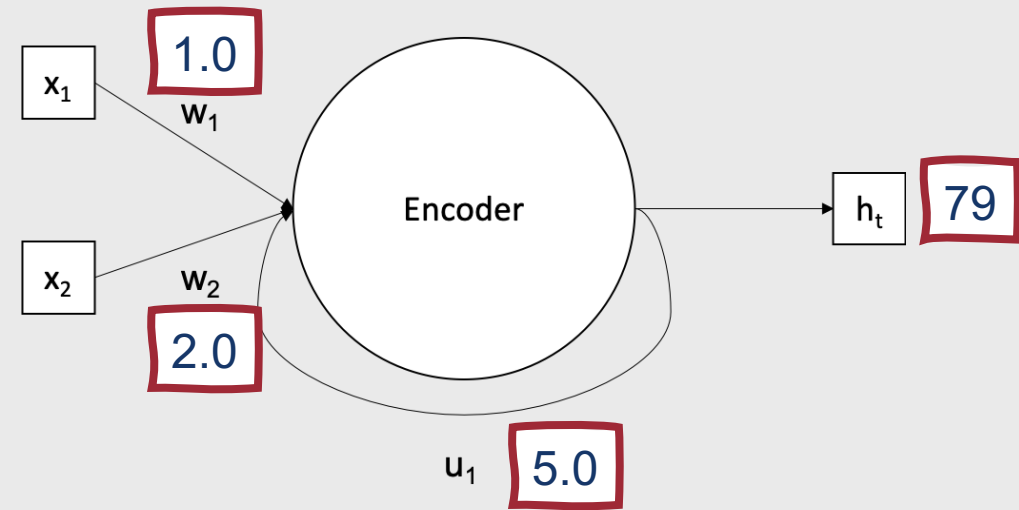
$$h_1 = \text{ReLU}((2 * 1.0 + 0 * 2.0) + 0 * 5.0) = \text{ReLU}(2) = 2$$

$$h_2 = \text{ReLU}((1 * 1.0 + 2 * 2.0) + 2 * 5.0) = \text{ReLU}(15) = 15$$

$$h_3 = \text{ReLU}((2 * 1.0 + 1 * 2.0) + 15 * 5.0) = \text{ReLU}(79) = 79$$

# Problem-Solving Questions

Assume we have an extremely simple encoder-decoder model, designed to translate a sarcastic sentence into its non-sarcastic counterpart. The encoder (pictured below) is comprised of a single RNN unit. Input tokens are represented as two-dimensional word embeddings, for which the encoder has learned weights  $w = [1.0, 2.0]$ . The encoder has also learned to weight the output of the RNN at the previous timestep as  $u = [5.0]$ . The unit uses a ReLU activation function.



The following word embedding dictionary is available:

word	$x_1$	$x_2$
i	2	0
loooooove	1	2
midterms	2	1

Encode the sequence below:

*i loooooove midterms*

$$h_1 = \text{ReLU}((2 * 1.0 + 0 * 2.0) + 0 * 5.0) = \text{ReLU}(2) = 2$$

$$h_2 = \text{ReLU}((1 * 1.0 + 2 * 2.0) + 2 * 5.0) = \text{ReLU}(15) = 15$$

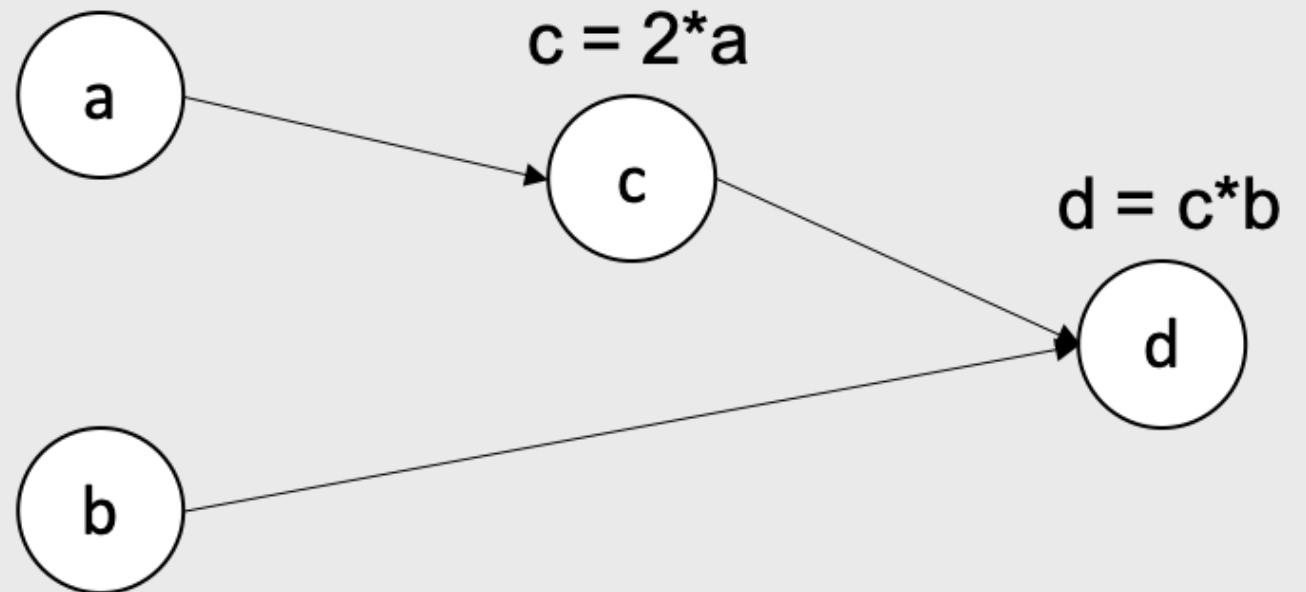
$$h_3 = \text{ReLU}((2 * 1.0 + 1 * 2.0) + 15 * 5.0) = \text{ReLU}(79) = 79$$





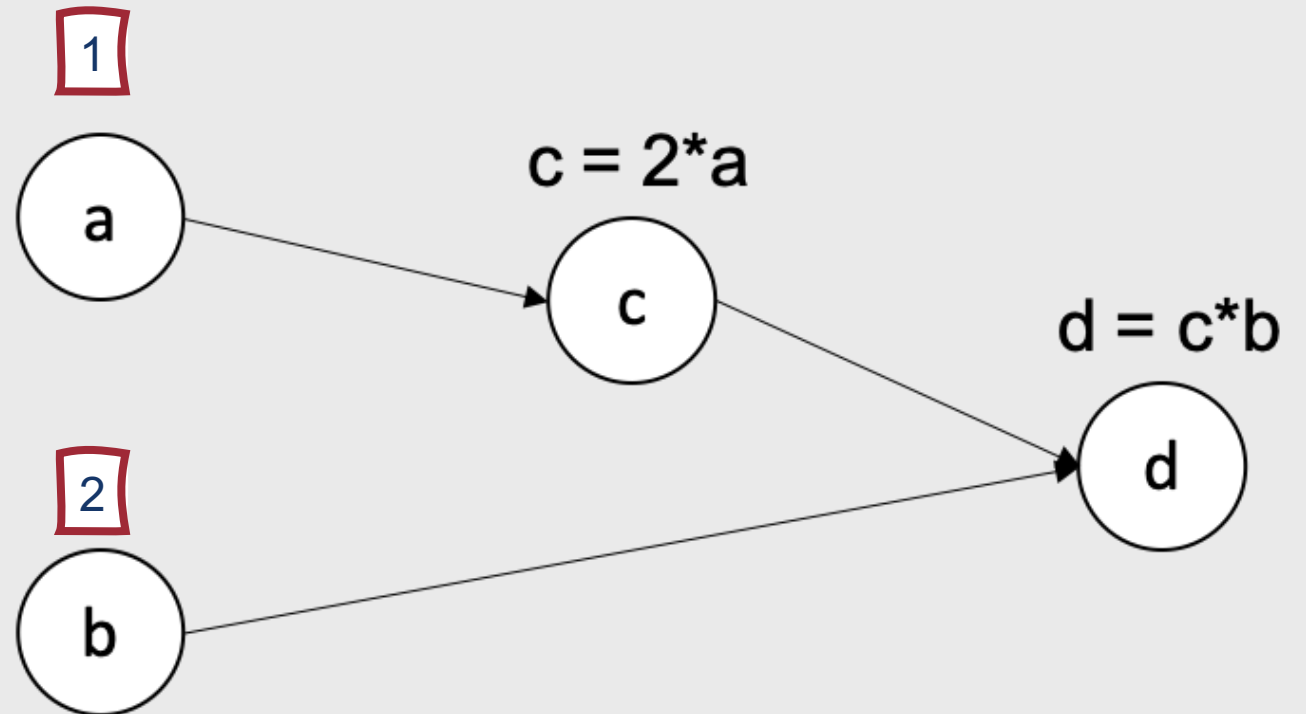
# Problem-Solving Questions

Given the computation graph below and the initial weights  $a = 1$  and  $b = 2$ , compute (a) a forward pass, and (b) a backward pass, to find the updated weights for  $a$  and  $b$ .



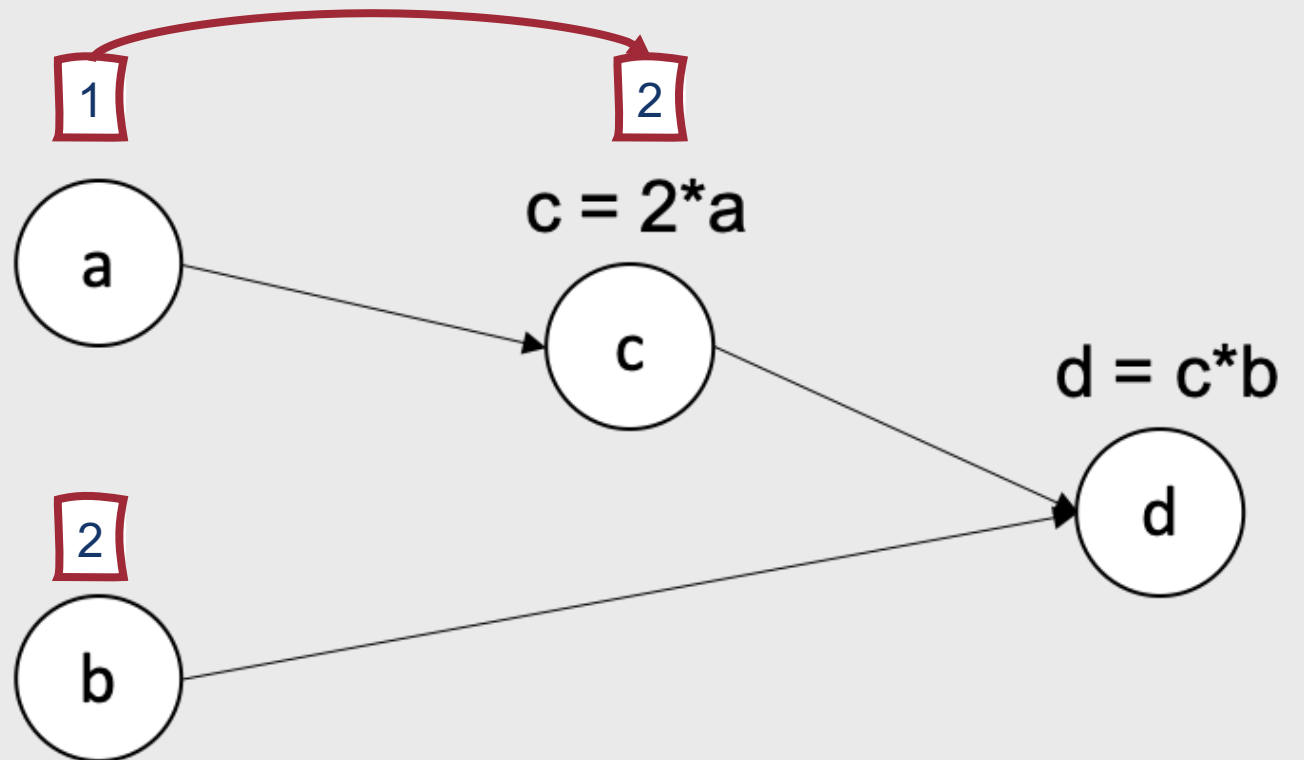
# Problem-Solving Questions

Given the computation graph below and the initial weights  $a = 1$  and  $b = 2$ , compute (a) a forward pass, and (b) a backward pass, to find the updated weights for  $a$  and  $b$ .



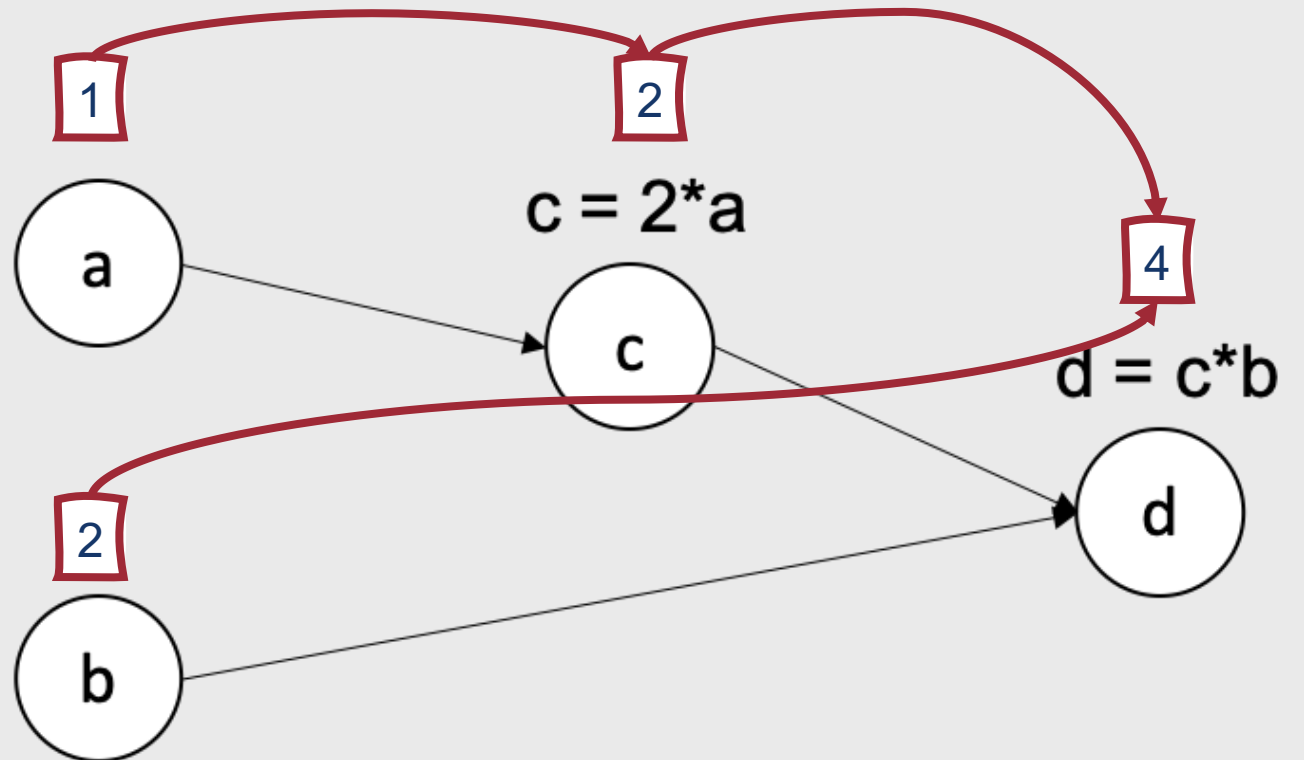
# Problem-Solving Questions

Given the computation graph below and the initial weights  $a = 1$  and  $b = 2$ , compute (a) a forward pass, and (b) a backward pass, to find the updated weights for  $a$  and  $b$ .



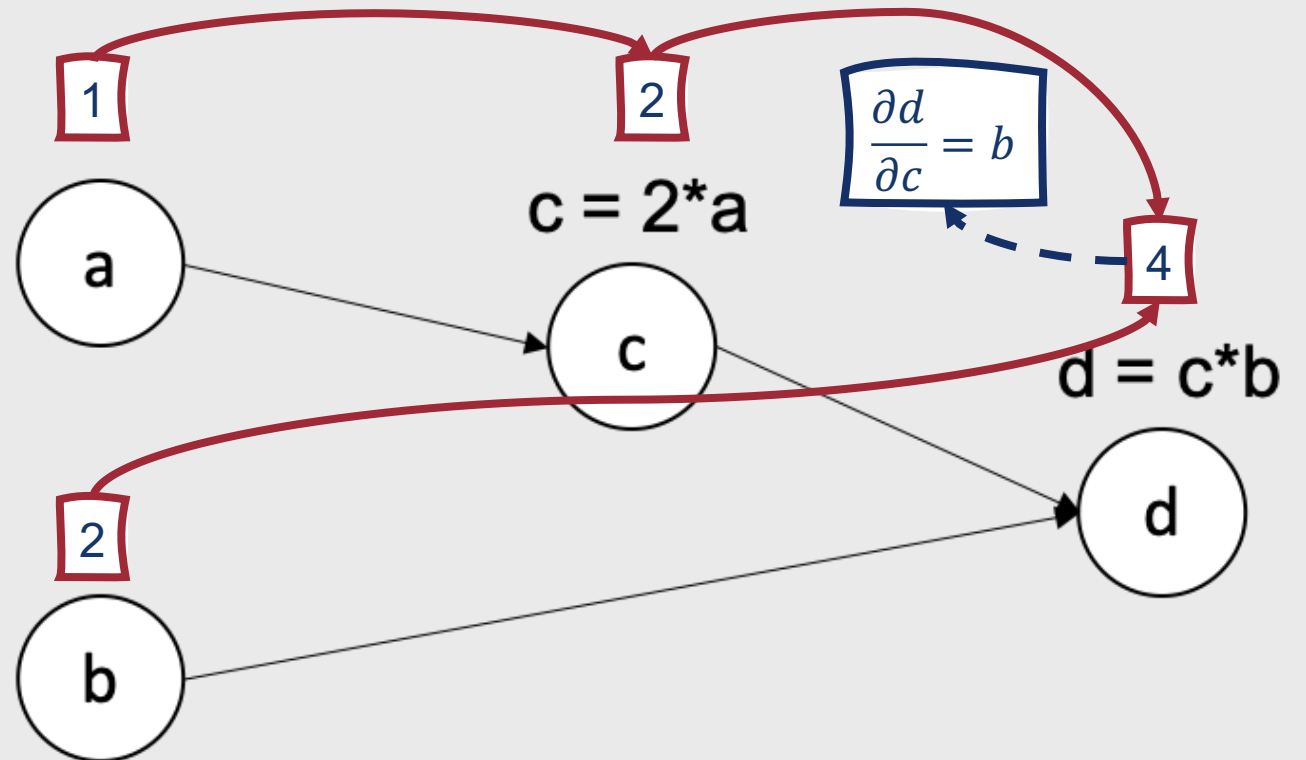
# Problem-Solving Questions

Given the computation graph below and the initial weights  $a = 1$  and  $b = 2$ , compute (a) a forward pass, and (b) a backward pass, to find the updated weights for  $a$  and  $b$ .



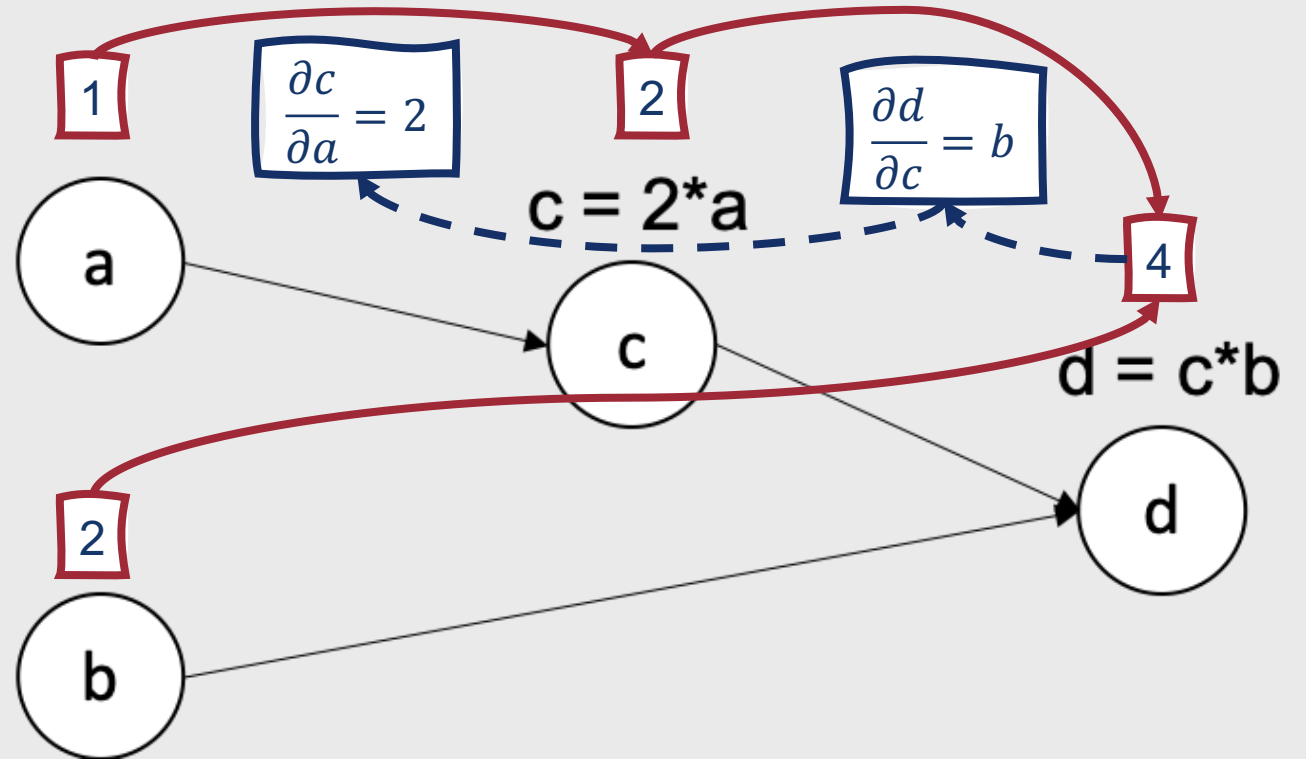
# Problem-Solving Questions

Given the computation graph below and the initial weights  $a = 1$  and  $b = 2$ , compute (a) a forward pass, and (b) a backward pass, to find the updated weights for  $a$  and  $b$ .



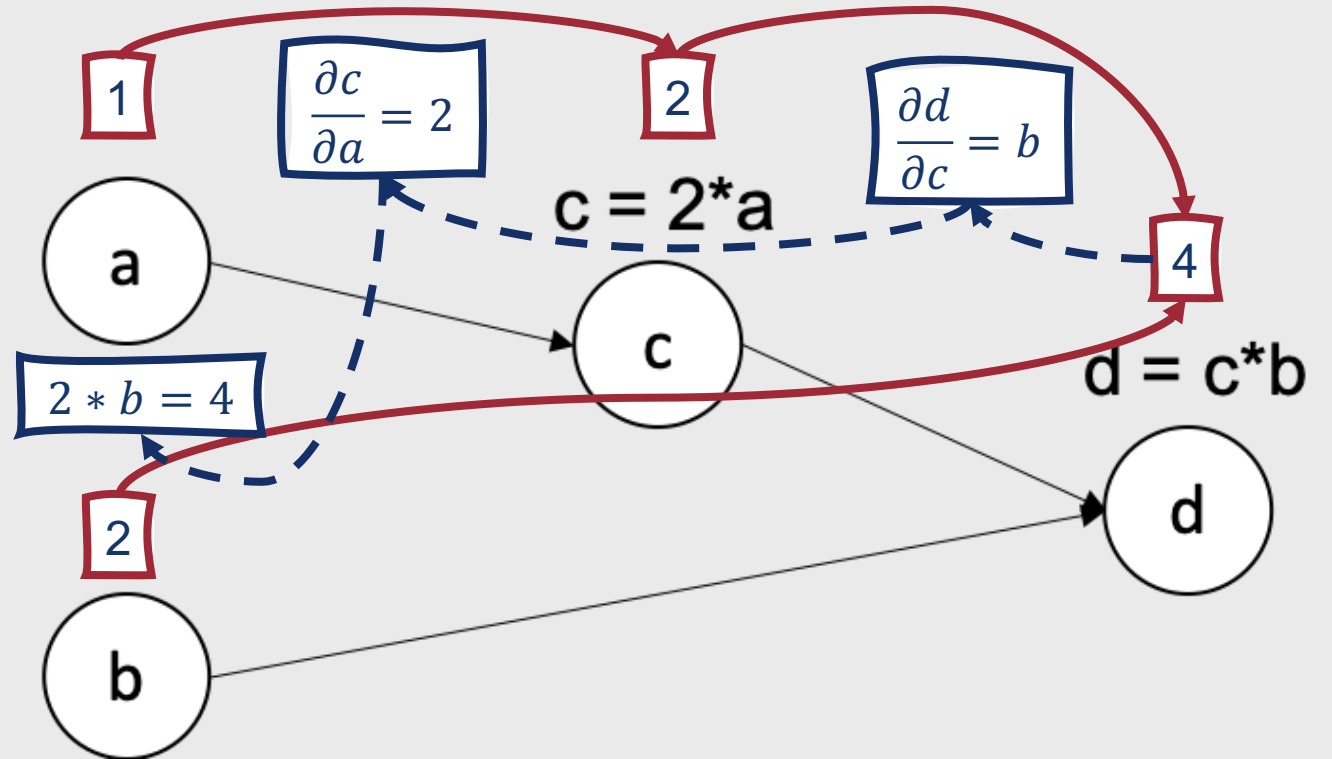
# Problem-Solving Questions

Given the computation graph below and the initial weights  $a = 1$  and  $b = 2$ , compute (a) a forward pass, and (b) a backward pass, to find the updated weights for  $a$  and  $b$ .



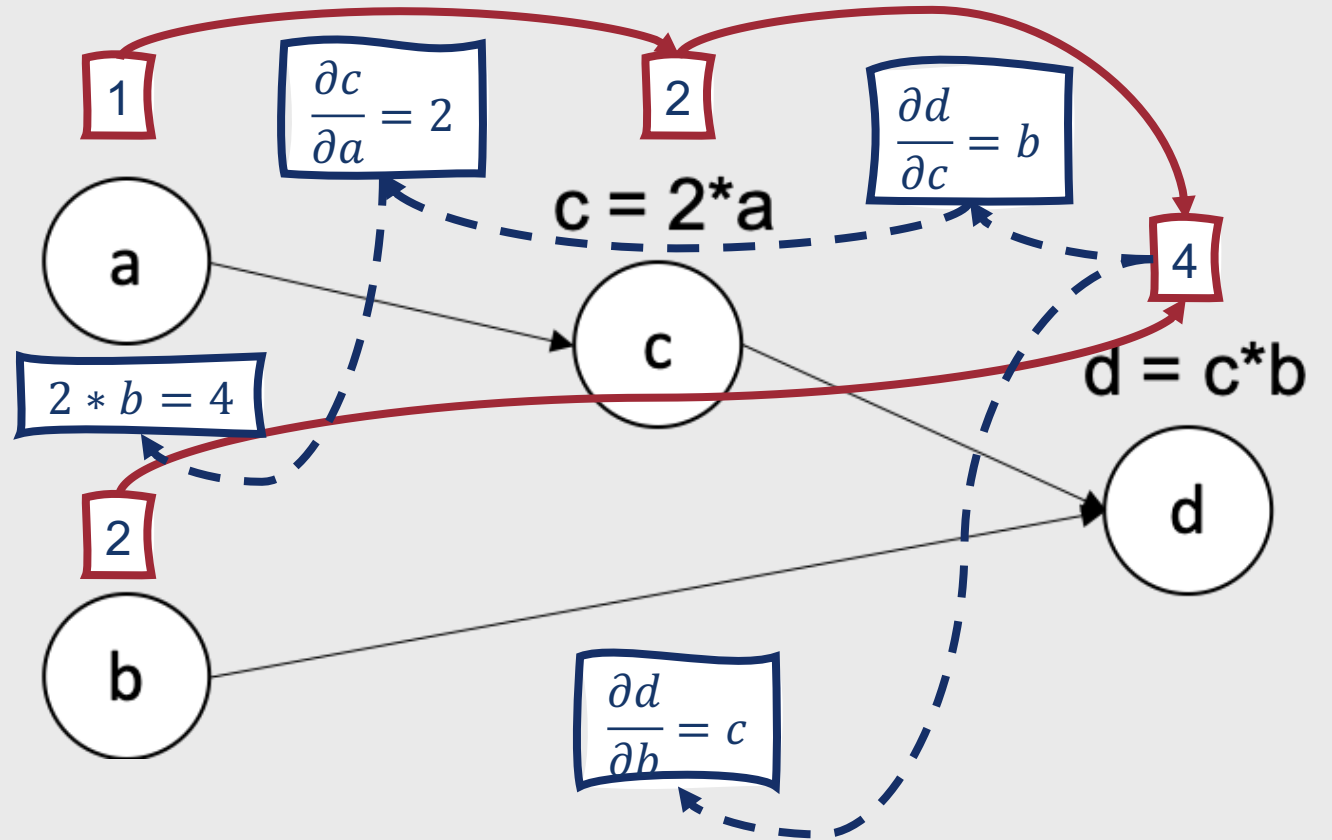
# Problem-Solving Questions

Given the computation graph below and the initial weights  $a = 1$  and  $b = 2$ , compute (a) a forward pass, and (b) a backward pass, to find the updated weights for  $a$  and  $b$ .



# Problem-Solving Questions

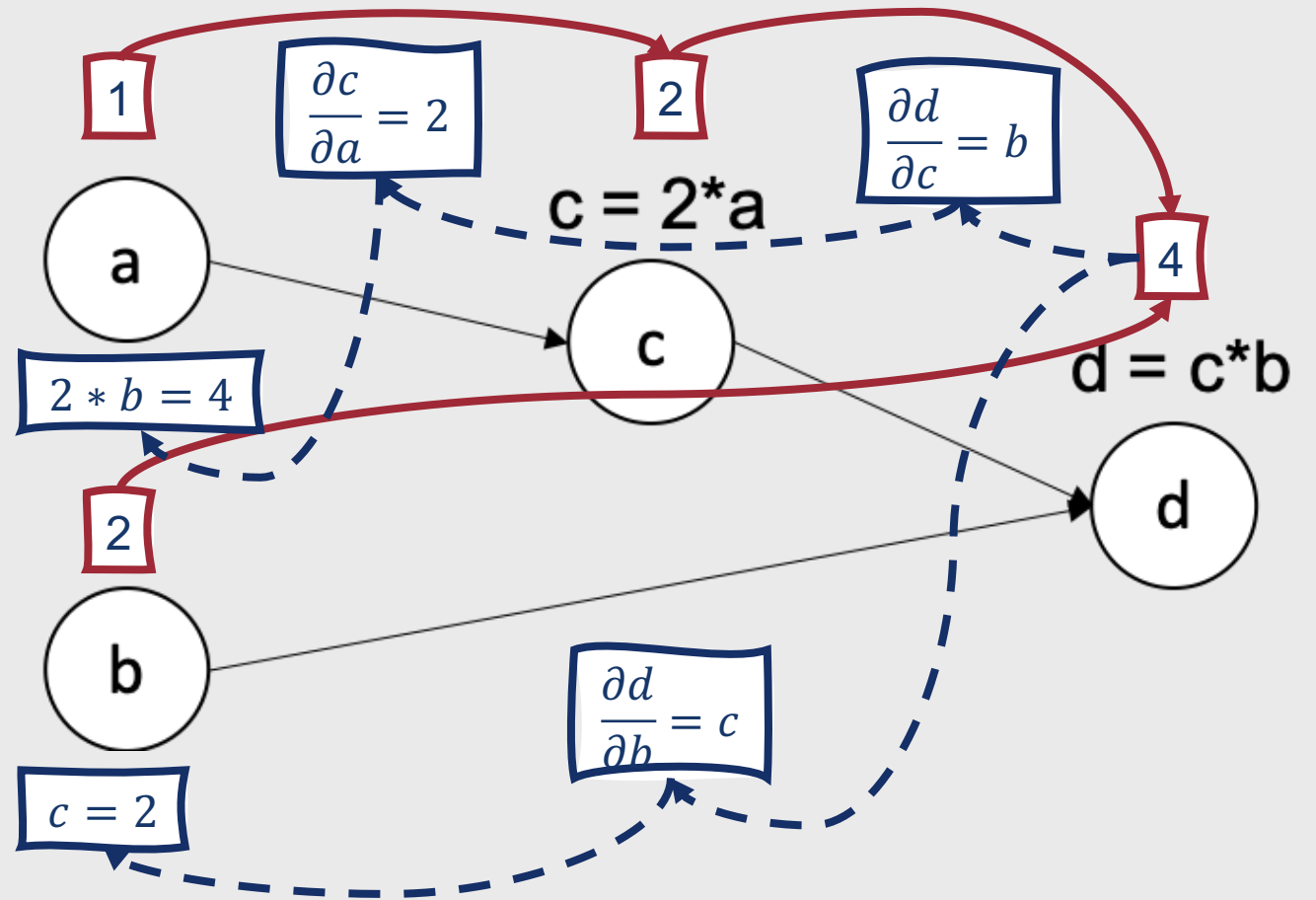
Given the computation graph below and the initial weights  $a = 1$  and  $b = 2$ , compute (a) a forward pass, and (b) a backward pass, to find the updated weights for  $a$  and  $b$ .





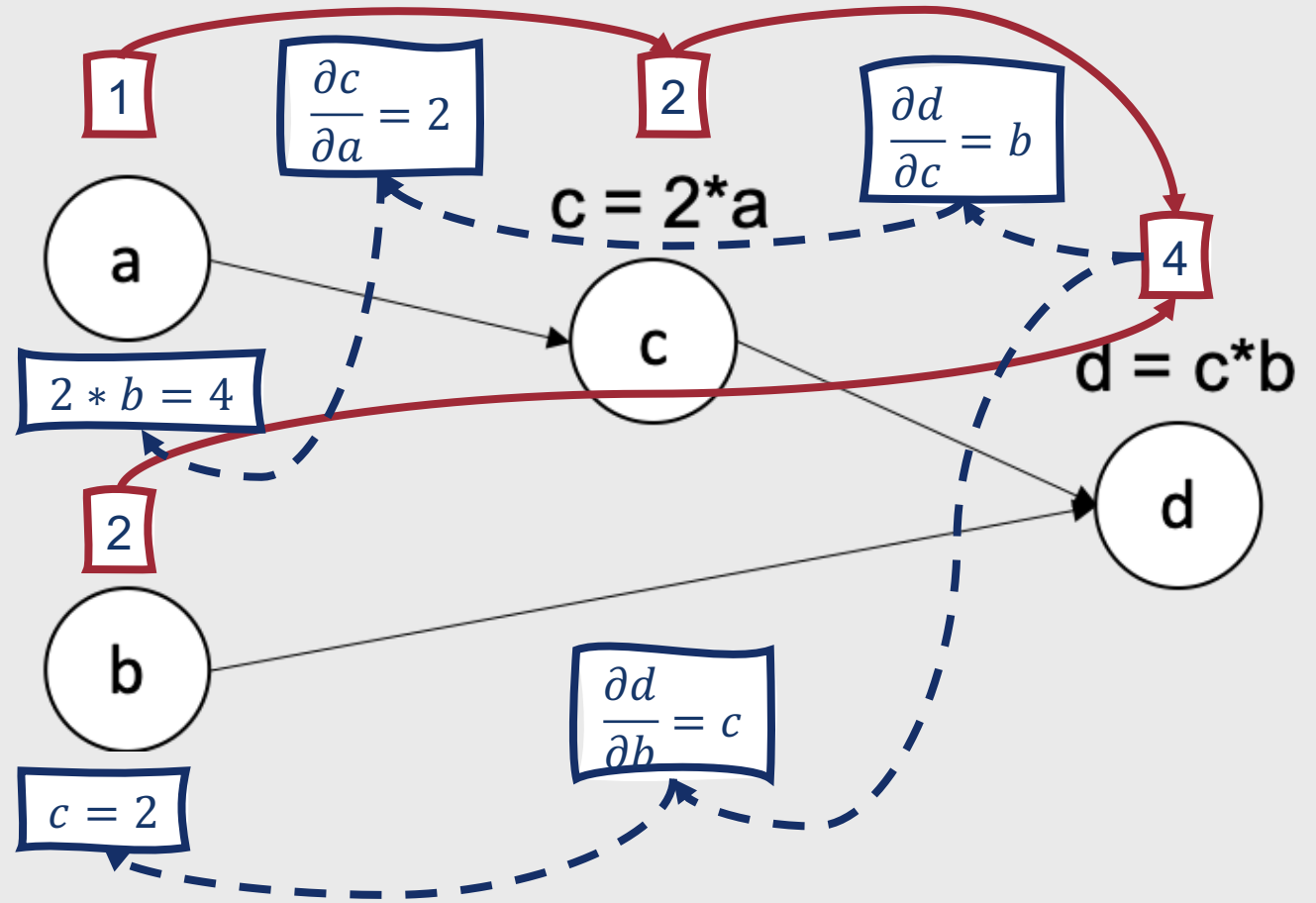
# Problem-Solving Questions

Given the computation graph below and the initial weights  $a = 1$  and  $b = 2$ , compute (a) a forward pass, and (b) a backward pass, to find the updated weights for  $a$  and  $b$ .



# Problem-Solving Questions

Given the computation graph below and the initial weights  $a = 1$  and  $b = 2$ , compute (a) a forward pass, and (b) a backward pass, to find the updated weights for  $a$  and  $b$ .



# Problem-Solving Questions

Build an entity grid for the following sentences, and compute the probability for the local entity transition  $\{s, -\}$ :

1. Natalie taught a class at UIC.
2. She liked UIC.
3. UIC liked brutalism.

# Problem-Solving Questions

Build an entity grid for the following sentences, and compute the probability for the local entity transition  $\{s, -\}$ :

1. Natalie taught a class at UIC.
2. She liked UIC.
3. UIC liked brutality.

Natalie taught a class at UIC.

She liked UIC.

UIC liked brutality.

	Natalie	class	UIC	brutality
S1				
S2				
S3				

# Problem-Solving Questions

Build an entity grid for the following sentences, and compute the probability for the local entity transition  $\{s, -\}$ :

1. Natalie taught a class at UIC.
2. She liked UIC.
3. UIC liked brutality.

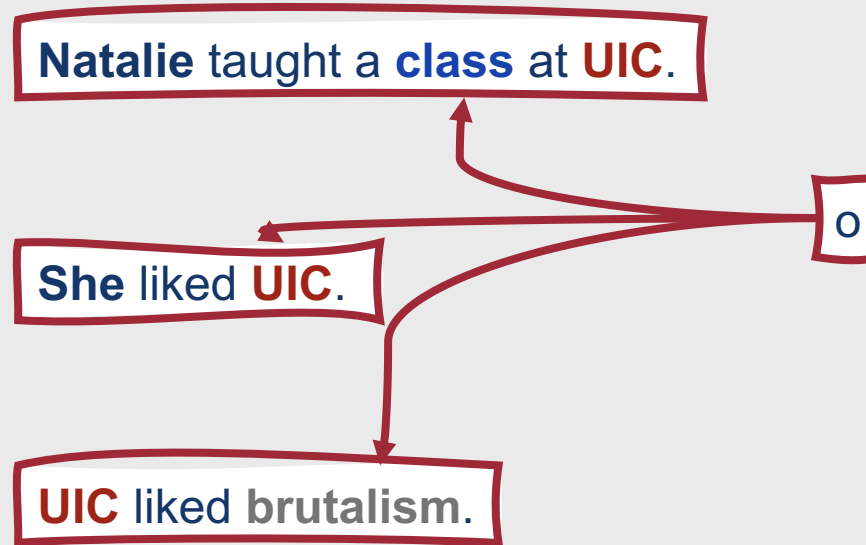


	Natalie	class	UIC	brutalism
S1	s			
S2	s			
S3			s	

## Problem-Solving Questions

Build an entity grid for the following sentences, and compute the probability for the local entity transition  $\{s, -\}$ :

1. Natalie taught a class at UIC.
2. She liked UIC.
3. UIC liked brutality.



	Natalie	class	UIC	brutalism
S1	s	o		
S2	s		o	
S3			s	o

# Problem-Solving Questions

Build an entity grid for the following sentences, and compute the probability for the local entity transition  $\{s, -\}$ :

1. Natalie taught a class at UIC.
2. She liked UIC.
3. UIC liked brutality.

Natalie taught a class at UIC.

She liked UIC.

UIC liked brutality.

X

	Natalie	class	UIC	brutalism
S1	s	o	x	
S2	s		o	
S3			s	o

# Problem-Solving Questions

Build an entity grid for the following sentences, and compute the probability for the local entity transition  $\{s, -\}$ :

1. Natalie taught a class at UIC.
2. She liked UIC.
3. UIC liked brutality.

Natalie taught a class at UIC.

She liked UIC.

UIC liked brutality.

	Natalie	class	UIC	brutality
S1	s	o	x	-
S2	s	-	o	-
S3	-	-	s	o



# Problem-Solving Questions

Build an entity grid for the following sentences, and compute the probability for the local entity transition  $\{s, -\}$ :

1. Natalie taught a class at UIC.
2. She liked UIC.
3. UIC liked brutality.

Natalie taught a class at UIC.

She liked UIC.

UIC liked brutality.

	Natalie	class	UIC	brutality
S1	s	o	x	-
S2	s	-	o	-
S3	-	-	s	o

$$p(\{s, -\}) = \frac{1}{8}$$

# Problem-Solving Questions

Build an entity grid for the following sentences, and compute the probability for the local entity transition  $\{s, -\}$ :

1. Natalie taught a class at UIC.
2. She liked UIC.
3. UIC liked brutality.

Natalie taught a class at UIC.

She liked UIC.

UIC liked brutality.

	Natalie	class	UIC	brutality
S1	s	o	x	-
S2	s	-	o	-
S3	-	-	s	o

$$p(\{s, -\}) = \frac{1}{8}$$



# Solution Time!



Bonus Question

## Bonus Question

Imagine that you are building a simple chatbot designed to tell people the weather when they ask. To get started, you pair up some of your friends and record them while they ask each other what the weather is, with the idea being that you will model your dialogue manager based on these conversations. You end up with the following sample transcripts.

Build a dialogue act labeling scheme based on the sample transcripts.

p1	hey		
p2	hi		
p1	what's the weather?		
p2	25 degrees fahrenheit		
p1	thanks	p1	what is the weather outside?
		p2	0 degrees celsius
		p1	okay, bye
p1	hello		
p2	howdy		
p1	what's the weather like?		
p2	huh?		
p1	What temperature is it right now?		
p2	30 degrees fahrenheit		
p1	cool, bye		

## Bonus Question

Imagine that you are building a simple chatbot designed to tell people the weather when they ask. To get started, you pair up some of your friends and record them while they ask each other what the weather is, with the idea being that you will model your dialogue manager based on these conversations. You end up with the following sample transcripts.

Build a dialogue act labeling scheme based on the sample transcripts.

p1	hey
p2	hi
p1	what's the weather?
p2	25 degrees fahrenheit
p1	thanks

p1	what is the weather outside?
p2	0 degrees celsius
p1	okay, bye

p1	hello
p2	howdy
p1	what's the weather like?
p2	huh?
p1	What temperature is it right now?
p2	30 degrees fahrenheit
p1	cool, bye

Dialogue Act	Description	Example
Greeting	Signals the beginning of a conversation	"hello"

## Bonus Question

Imagine that you are building a simple chatbot designed to tell people the weather when they ask. To get started, you pair up some of your friends and record them while they ask each other what the weather is, with the idea being that you will model your dialogue manager based on these conversations. You end up with the following sample transcripts.

Build a dialogue act labeling scheme based on the sample transcripts.

p1	hey
p2	hi
p1	what's the weather?
p2	25 degrees fahrenheit
p1	thanks

p1	what is the weather outside?
p2	0 degrees celsius
p1	okay, bye

p1	hello
p2	howdy
p1	what's the weather like?
p2	huh?
p1	What temperature is it right now?
p2	30 degrees fahrenheit
p1	cool, bye

Dialogue Act	Description	Example
Greeting	Signals the beginning of a conversation	"hello"
Weather_Question	Asks for the current weather	"what's the weather?"

## Bonus Question

Imagine that you are building a simple chatbot designed to tell people the weather when they ask. To get started, you pair up some of your friends and record them while they ask each other what the weather is, with the idea being that you will model your dialogue manager based on these conversations. You end up with the following sample transcripts.

Build a dialogue act labeling scheme based on the sample transcripts.

p1	hey
p2	hi
p1	what's the weather?
p2	25 degrees fahrenheit
p1	thanks

p1	what is the weather outside?
p2	0 degrees celsius
p1	okay, bye

p1	hello
p2	howdy
p1	what's the weather like?
p2	huh?
p1	What temperature is it right now?
p2	30 degrees fahrenheit
p1	cool, bye

Dialogue Act	Description	Example
Greeting	Signals the beginning of a conversation	"hello"
Weather_Question	Asks for the current weather	"what's the weather?"
Weather_Response	Provides the current weather	"it's -5 degrees celsius"

## Bonus Question

Imagine that you are building a simple chatbot designed to tell people the weather when they ask. To get started, you pair up some of your friends and record them while they ask each other what the weather is, with the idea being that you will model your dialogue manager based on these conversations. You end up with the following sample transcripts.

Build a dialogue act labeling scheme based on the sample transcripts.

p1	hey
p2	hi
p1	what's the weather?
p2	25 degrees fahrenheit
p1	thanks

p1	what is the weather outside?
p2	0 degrees celsius
p1	okay, bye

p1	hello
p2	howdy
p1	what's the weather like?
p2	huh?
p1	What temperature is it right now?
p2	30 degrees fahrenheit
p1	cool, bye

Dialogue Act	Description	Example
Greeting	Signals the beginning of a conversation	"hello"
Weather_Question	Asks for the current weather	"what's the weather?"
Weather_Response	Provides the current weather	"it's -5 degrees celsius"
Clarifying_Question	Asks for clarification	"what did you say?"



## Bonus Question

Imagine that you are building a simple chatbot designed to tell people the weather when they ask. To get started, you pair up some of your friends and record them while they ask each other what the weather is, with the idea being that you will model your dialogue manager based on these conversations. You end up with the following sample transcripts.

Build a dialogue act labeling scheme based on the sample transcripts.

p1	hey
p2	hi
p1	what's the weather?
p2	25 degrees fahrenheit
p1	thanks

p1	what is the weather outside?
p2	0 degrees celsius
p1	okay, bye

p1	hello
p2	howdy
p1	what's the weather like?
p2	huh?
p1	What temperature is it right now?
p2	30 degrees fahrenheit
p1	cool, bye

Dialogue Act	Description	Example
Greeting	Signals the beginning of a conversation	"hello"
Weather_Question	Asks for the current weather	"what's the weather?"
Weather_Response	Provides the current weather	"it's -5 degrees celsius"
Clarifying_Question	Asks for clarification	"what did you say?"
Acknowledgement	Acknowledges the previous utterance	"okay"

## Bonus Question

Imagine that you are building a simple chatbot designed to tell people the weather when they ask. To get started, you pair up some of your friends and record them while they ask each other what the weather is, with the idea being that you will model your dialogue manager based on these conversations. You end up with the following sample transcripts.

Build a dialogue act labeling scheme based on the sample transcripts.

p1	hey
p2	hi
p1	what's the weather?
p2	25 degrees fahrenheit
p1	thanks

p1	what is the weather outside?
p2	0 degrees celsius
p1	okay, bye

p1	hello
p2	howdy
p1	what's the weather like?
p2	huh?
p1	What temperature is it right now?
p2	30 degrees fahrenheit
p1	cool, bye

Dialogue Act	Description	Example
Greeting	Signals the beginning of a conversation	"hello"
Weather_Question	Asks for the current weather	"what's the weather?"
Weather_Response	Provides the current weather	"it's -5 degrees celsius"
Clarifying_Question	Asks for clarification	"what did you say?"
Acknowledgement	Acknowledges the previous utterance	"okay"
Goodbye	Signals the end of a conversation	"bye"

## Bonus Question

Imagine that you are building a simple chatbot designed to tell people the weather when they ask. To get started, you pair up some of your friends and record them while they ask each other what the weather is, with the idea being that you will model your dialogue manager based on these conversations. You end up with the following sample transcripts.

Build a dialogue act labeling scheme based on the sample transcripts.

p1	hey
p2	hi
p1	what's the weather?
p2	25 degrees fahrenheit
p1	thanks

p1	what is the weather outside?
p2	0 degrees celsius
p1	okay, bye

p1	hello
p2	howdy
p1	what's the weather like?
p2	huh?
p1	What temperature is it right now?
p2	30 degrees fahrenheit
p1	cool, bye

Dialogue Act	Description	Example
Greeting	Signals the beginning of a conversation	"hello"
Weather_Question	Asks for the current weather	"what's the weather?"
Weather_Response	Provides the current weather	"it's -5 degrees celsius"
Clarifying_Question	Asks for clarification	"what did you say?"
Acknowledgement	Acknowledges the previous utterance	"okay"
Goodbye	Signals the end of a conversation	"bye"





Exam Location: Same classroom  
as always (TBH 180B)



Exam Time: Same time as class  
(Tuesday from 9:30-10:45 a.m.)

# A few remaining details....



**Good luck!**