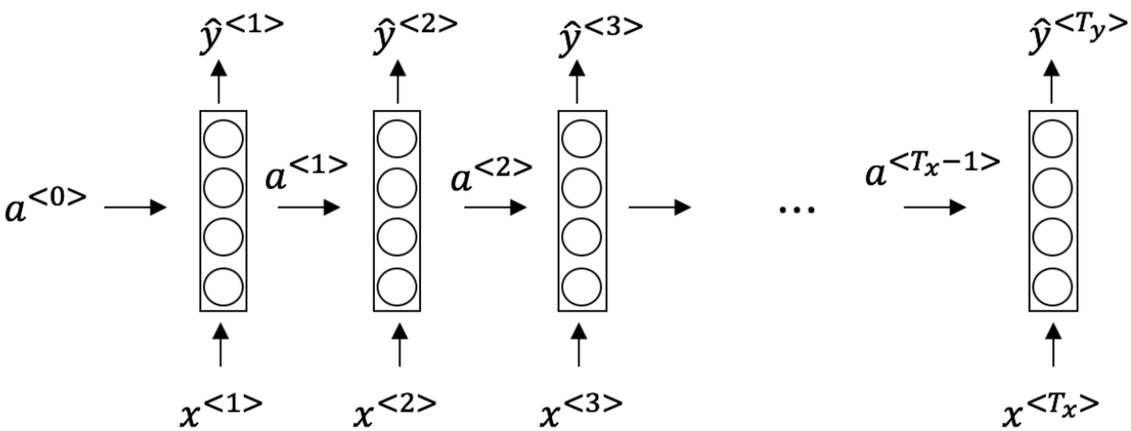
1.Suppose your training examples are sentences (sequences of words). Which of the following refers to the j^{th} word in the i^{th} training example? 1 point

- ○ $x^{(i)<j>}$
- ○ $x^{<i>(j)}$
- ○ $x^{(j)<i>}$
- ○ $x^{<j>(i)}$

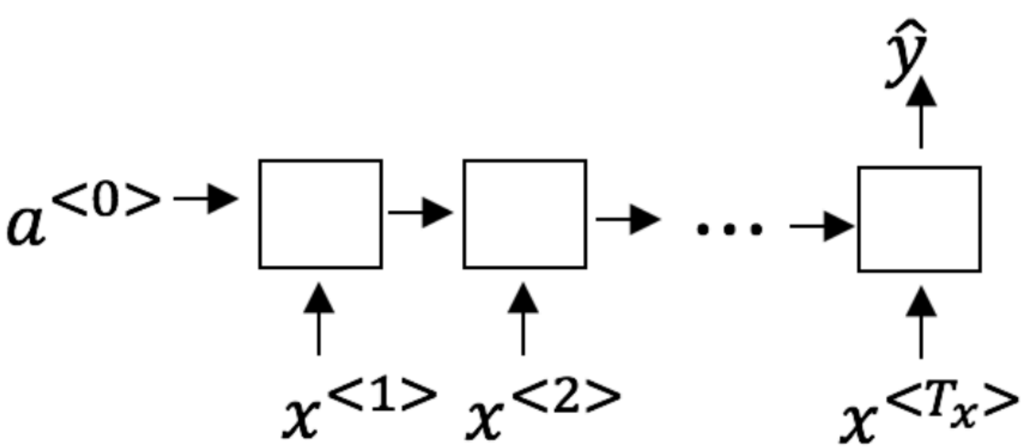
2.Consider this RNN: 1 point



True/False: This specific type of architecture is appropriate when Tx>Ty

- ○ True
- ○ False

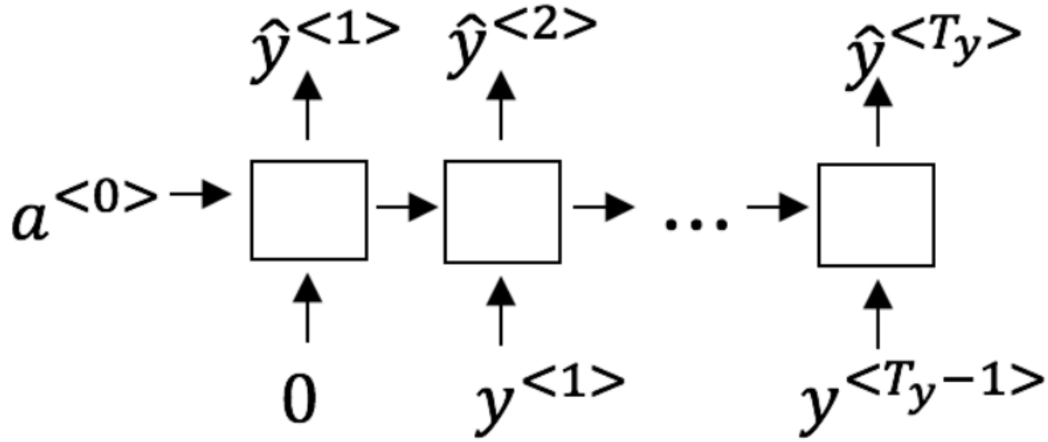
3.To which of these tasks would you apply a many-to-one RNN architecture? (Check all that apply). 1 point



- ☐ Speech recognition (input an audio clip and output a transcript)
- ☐ Sentiment classification (input a piece of text and output a 0/1 to denote positive or negative sentiment)

- [ ] Image classification (input an image and output a label)
- [ ] Gender recognition from speech (input an audio clip and output a label indicating the speaker's gender)

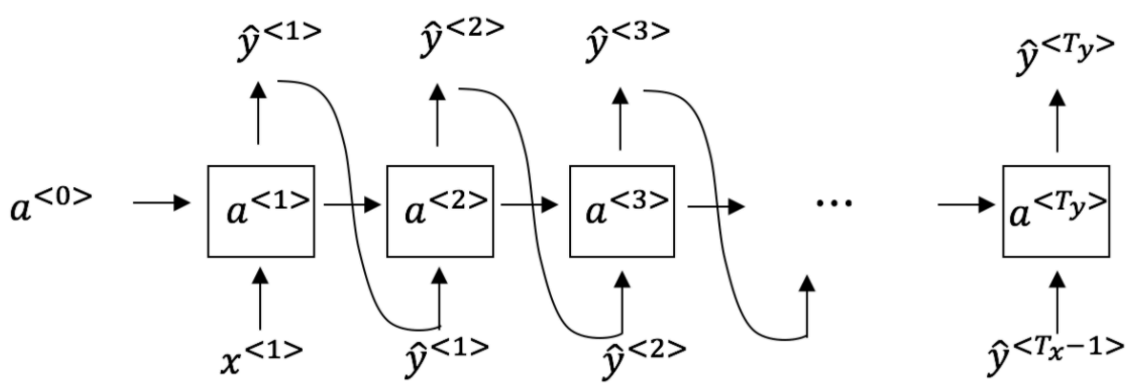
4.Using this as the training model below, answer the following: 1 point



True/False: At the t^{th} time step the RNN is estimating $P(y^{<t>} \mid y^{<1>}, y^{<2>}, \ldots, y^{<t-1>})$

- ( ) True
- ( ) False

5.You have finished training a language model RNN and are using it to sample random sentences, as follows: 1 point



True/False: In this sample sentence, step t uses the probabilities output by the RNN to pick the highest probability word for that time-step. Then it passes the ground-truth word from the training set to the next time-step.

- ( ) True
- ( ) False

6.You are training an RNN model, and find that your weights and activations are all taking on the value of NaN ("Not a Number"). Which of these is the most likely cause of this problem? 1 point

- ( ) Vanishing gradient problem.

- Exploding gradient problem.
- The model used the ReLU activation function to compute g(z), where z is too large.
- The model used the Sigmoid activation function to compute g(z), where z is too large.

7.Suppose you are training an LSTM. You have an 80000 word vocabulary, and are using an LSTM with 800-dimensional activations $a^{<t>}$. What is the dimension of Γ_u at each time step? 1 point

- 8
- 100
- 80000
- 800

8.Sarah proposes to simplify the GRU by always removing the Γu. I.e., setting Γu = 0. Ashely proposes to simplify the GRU by removing the Γr. I. e., setting Γr= 1 always. Which of these models is more likely to work without vanishing gradient problems even when trained on very long input sequences? 1 point

GRU

$$\tilde{c}^{<t>} = \tanh(W_c[\Gamma_r * c^{<t-1>}, x^{<t>}] + b_c)$$

$$\Gamma_u = \sigma(W_u[\, c^{<t-1>}, x^{<t>}] + b_u)$$

$$\Gamma_r = \sigma(W_r[\, c^{<t-1>}, x^{<t>}] + b_r)$$

$$c^{<t>} = \Gamma_u * \tilde{c}^{<t>} + (1 - \Gamma_u) * c^{<t-1>}$$

$$a^{<t>} = c^{<t>}$$

- Sarah’s model (removing Γu), because if Γr≈1 for a timestep, the gradient can propagate back through that timestep without much decay.
- Ashely’s model (removing Γr), because if Γu≈0 for a timestep, the gradient can propagate back through that timestep without much decay.
- Sarah’s model (removing Γu), because if Γr ≈0 for a timestep, the gradient can propagate back through that timestep without much decay.
- Ashely’s model (removing Γr), because if Γu≈1 for a timestep, the gradient can propagate back through that timestep without much decay.

9.Here are the equations for the GRU and the LSTM: 1 point

GRU	LSTM
$\tilde{c}^{<t>} = \tanh(W_c[\Gamma_r * c^{<t-1>}, x^{<t>}] + b_c)$	$\tilde{c}^{<t>} = \tanh(W_c[a^{<t-1>}, x^{<t>}] + b_c)$
$\Gamma_u = \sigma(W_u[\, c^{<t-1>}, x^{<t>}] + b_u)$	$\Gamma_u = \sigma(W_u[\, a^{<t-1>}, x^{<t>}] + b_u)$
$\Gamma_r = \sigma(W_r[\, c^{<t-1>}, x^{<t>}] + b_r)$	$\Gamma_f = \sigma(W_f[\, a^{<t-1>}, x^{<t>}] + b_f)$
$c^{<t>} = \Gamma_u * \tilde{c}^{<t>} + (1 - \Gamma_u) * c^{<t-1>}$	$\Gamma_o = \sigma(W_o[\, a^{<t-1>}, x^{<t>}] + b_o)$
$a^{<t>} = c^{<t>}$	$c^{<t>} = \Gamma_u * \tilde{c}^{<t>} + \Gamma_f * c^{<t-1>}$
	$a^{<t>} = \Gamma_o * \tanh c^{<t>}$

From these, we can see that the Update Gate and Forget Gate in the LSTM play a role similar to _______ and ______ in the GRU. What should go in the blanks?

- ○ Γ_u and $1 - \Gamma_u$
- ○ Γ_u and Γ_r
- ○ $1 - \Gamma_u$ and Γ_u
- ○ Γ_r and Γ_u

10. You have a pet dog whose mood is heavily dependent on the current and past few days' weather. You've collected data for the past 365 days on the weather, which you represent as a sequence as $x^{<1>}, \ldots, x^{<365>}$. You've also collected data on your dog's mood, which you represent as $y^{<1>}, \ldots, y^{<365>}$. You'd like to build a model to map from $x \rightarrow y$. Should you use a Unidirectional RNN or Bidirectional RNN for this problem? 1 point

- ○ Bidirectional RNN, because this allows the prediction of mood on day t to take into account more information.
- ○ Bidirectional RNN, because this allows backpropagation to compute more accurate gradients.
- ○ Unidirectional RNN, because the value of $y^{<t>}$ depends only on $x^{<1>}, \ldots, x^{<t>}$, but not on $x^{<t+1>}, \ldots, x^{<365>}$
- ○ Unidirectional RNN, because the value of $y^{<t>}$ depends only on $x^{<t>}$, and not other days' weather.