Congratulations! You passed!

Grade received 90% **Latest Submission Grade** 90% **To pass** 80% or higher

Go to next item

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

1 / 1 point

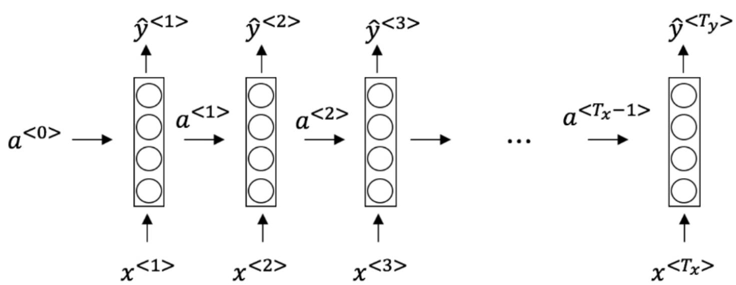
- ○ $x^{(l)<k>}$
- ◉ $x^{(k)<l>}$
- ○ $x^{<l>(k)}$
- ○ $x^{<k>(l)}$

Expand

✓ **Correct**
We index into the k^{th} row first to get to the k^{th} training example (represented by parentheses), then the l^{th} column to get to the l^{th} word (represented by the brackets).

2. Consider this RNN:

1 / 1 point



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

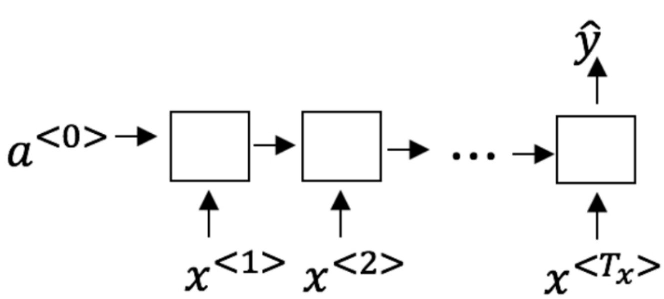
- ○ True
- ◉ False

Expand

✓ **Correct**
Correct! This type of architecture is for applications where the input and output sequence length is the same.

3. To which of these tasks would you apply a many-to-one RNN architecture?

1 / 1 point



- [ ] Image classification (input an image and output a label)
- [x] Music genre recognition

✓ **Correct**
This is an example of many-to-one architecture.

- [x] Language recognition from speech (input an audio clip and output a label indicating the language being spoken)

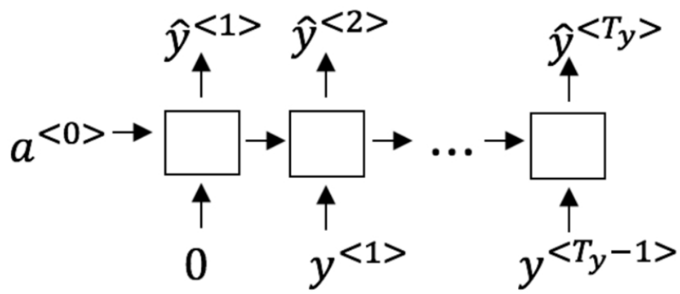
✓ **Correct**
This is an example of many-to-one architecture.

- [ ] Speech recognition (input an audio clip and output a transcript)

Expand

Correct
Great, you got all the right answers.

4. You are training this RNN language model. **1 / 1 point**



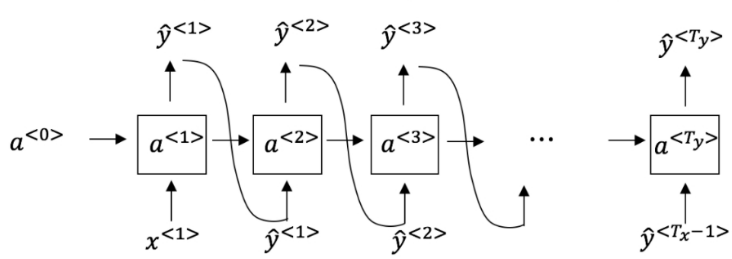
At the t^{th} time step, what is the RNN doing?

- ○ Estimating $P(y^{<1>}, y^{<2>}, \ldots, y^{<t-1>})$
- ○ Estimating $P(y^{<t>})$
- ◉ Estimating $P(y^{<t>} \mid y^{<1>}, y^{<2>}, \ldots, y^{<t-1>})$

Expand

Correct
Yes, in a language model we try to predict the next step based on the knowledge of all prior steps.

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

Expand

Correct
The probabilities output by the RNN are not used to pick the highest probability word and the ground-truth word from the training set is not the input to the next time-step.

6. True/False: If you are training an RNN model, and find that your weights and activations are all taking on the value of NaN ("Not a Number") then you have an exploding gradient problem.

0 / 1 point

- ◉ False
- ○ True

Expand

Incorrect
Incorrect! Exploding gradients happen when large error gradients accumulate and result in very large updates to the NN model weights during training. These weights can become too large and cause an overflow, identified as NaN.

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 / 1 point

- ○ 100
- ◉ 800
- ○ 8
- ○ 80000

Expand

Correct
Correct, Γ_u is a vector of dimension equal to the number of hidden units in the LSTM.

8. Here are the update equations for the GRU.

1 / 1 point

GRU

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

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

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

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

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

Alice proposes to simplify the GRU by always removing the Γ_u. I.e., setting $\Gamma_u = 0$. Betty proposes to simplify the GRU by removing the Γ_r. I. e., setting $\Gamma_r = 1$ always. Which of these models is more likely to work without vanishing gradient problems even when trained on very long input sequences?

- ○ Alice's model (removing Γ_u), because if $\Gamma_r \approx 0$ for a timestep, the gradient can propagate back through that timestep without much decay.
- ○ Alice's model (removing Γ_u), because if $\Gamma_r \approx 1$ for a timestep, the gradient can propagate back through that timestep without much decay.
- ◉ Betty's model (removing Γ_r), because if $\Gamma_u \approx 0$ for a timestep, the gradient can propagate back through that timestep without much decay.
- ○ Betty's model (removing Γ_r), because if $\Gamma_u \approx 1$ for a timestep, the gradient can propagate back through that timestep without much decay.

Expand

Correct

Yes. For the signal to backpropagate without vanishing, we need $c^{<t>}$ to be highly dependent on $c^{<t-1>}$.

9. True/False: Using the equations for the GRU and LSTM below the Update Gate and Forget Gate in the LSTM play a role similar to 1- Γu and Γu.

1 / 1 point

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

- True
- False (selected)

Expand

Correct

Instead of using Γu to compute 1 - Γu, LSTM uses 2 gates (Γu and Γf) to compute the final value of the hidden state. So, Γf is used instead of 1 - Γu.

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 / 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 $x^{<t+1>}, \ldots, x^{<365>}$ (selected)
- Unidirectional RNN, because the value of $y^{<t>}$ depends only on $x^{<t>}$, and not other days' weather.

Expand

Correct

Yes!