## Congratulations! You passed!

**Grade received** 80% **Latest Submission Grade** 80% **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 $s^{th}$ word in the $r^{th}$ training example?

1 / 1 point

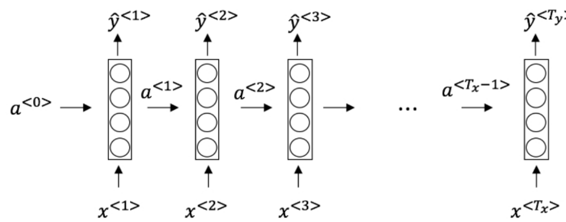
- [ ] $x^{<r>(s)}$
- [ ] $x^{(s)<r>}$
- [ ] $x^{<s>(r)}$
- [x] $x^{(r)<s>}$

Expand

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

**2.** Consider this RNN:

1 / 1 point




This specific type of architecture is appropriate when:

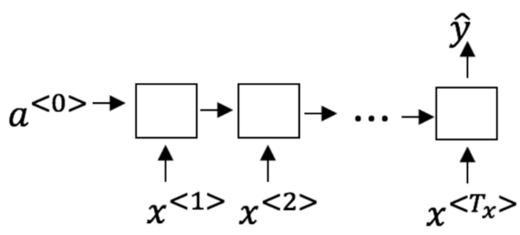
- [x] $T_x = T_y$
- [ ] $T_x < T_y$
- [ ] $T_x > T_y$
- [ ] $T_x = 1$

Expand

**Correct**
It is appropriate when every input should have an output.

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

0 / 1 point




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

**!** **This should not be selected**
This is an example of one-to-one architecture.

- [x] Music genre recognition

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

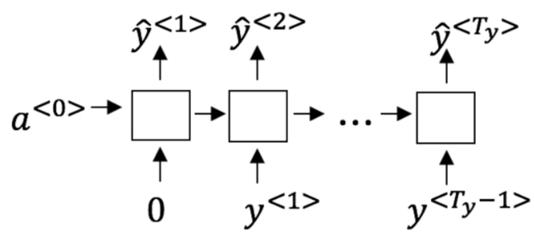
- [ ] Language recognition from speech (input an audio clip and output a label indicating the language being spoken)
- [ ] Speech recognition (input an audio clip and output a transcript)

Expand

**Incorrect**
You didn't select all the correct answers

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

1 / 1 point




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

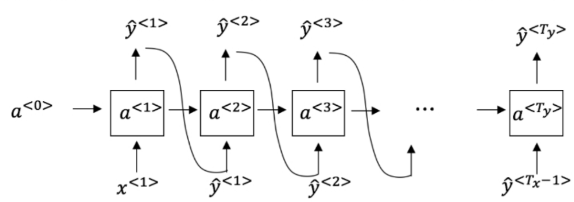
- ○ False
- ◉ True

Expand

**Correct**
Yes, in a training model we try to predict the next step based on 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




What are you doing at each time step $t$?

- ○ (i) Use the probabilities output by the RNN to pick the highest probability word for that time-step as $\hat{y}^{<t>}$. (ii) Then pass the ground-truth word from the training set to the next time-step.
- ○ (i) Use the probabilities output by the RNN to randomly sample a chosen word for that time-step as $\hat{y}^{<t>}$.(ii) Then pass the ground-truth word from the training set to the next time-step.
- ○ (i) Use the probabilities output by the RNN to pick the highest probability word for that time-step as $\hat{y}^{<t>}$.(ii) Then pass this selected word to the next time-step.
- ◉ (i) Use the probabilities output by the RNN to randomly sample a chosen word for that time-step as $\hat{y}^{<t>}$.(ii) Then pass this selected word to the next time-step.

Expand

**Correct**

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.

1 / 1 point

- ○ False
- ◉ True

Expand

**Correct**
Correct! 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 a 50000 word vocabulary, and are using an LSTM with 500-dimensional activations $a^{<t>}$. What is the dimension of $\Gamma_u$ at each time step?

1 / 1 point

- ○ 50000

- 5
- 500
- 200

Expand

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

**8.** True/False: In order to simplify the GRU without vanishing gradient problems even when training on very long sequences you should remove the $\Gamma_r$ i.e., setting $\Gamma_r = 1$ always.

1 / 1 point

- True
- False

Expand

Correct
If Γu≈0 for a timestep, the gradient can propagate back through that timestep without much decay. For the signal to backpropagate without vanishing, we need $c^{<t>}$ to be highly dependent on $c^{<t-1>}$.

**9.** Here are the equations for the GRU and the LSTM:

1 / 1 point

**GRU**

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

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

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

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

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

**LSTM**

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

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

$$\Gamma_f = \sigma(W_f[\, a^{<t-1>}, x^{<t>}] + b_f)$$

$$\Gamma_o = \sigma(W_o[\, a^{<t-1>}, x^{<t>}] + b_o)$$

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

$$a^{<t>} = \Gamma_o * \tanh c^{<t>}$$

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?

- $\Gamma_u$ and $1 - \Gamma_u$
- $\Gamma_u$ and $\Gamma_r$
- $1 - \Gamma_u$ and $\Gamma_u$
- $\Gamma_r$ and $\Gamma_u$

Expand

Correct
Yes, correct!

**10.** Your 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 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?

0 / 1 point

- Bidirectional RNN, because this allows backpropagation to compute more accurate gradients.
- Bidirectional RNN, because this allows the prediction of mood on day t to take into account more information.
- Unidirectional RNN, because the value of

  $y^{<t>}$

  depends only on

Loading [MathJax]/jax/output/CommonHTML/jax.js

Expand

Incorrect
Your mood is contingent on the current and past few days' weather, not on the current, past, AND future days' weather.