Congratulations! You passed!

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

Recurrent Neural Networks

Quiz • 30 min

Recurrent Neural Networks

Latest Submission Grade 90%

Due Apr 11, 2:59 PM CST **Attempts** 3 every 8 hours

Try again

Receive grade

To Pass 80% or higher

Your grade

90%

View Feedback

We keep your highest score

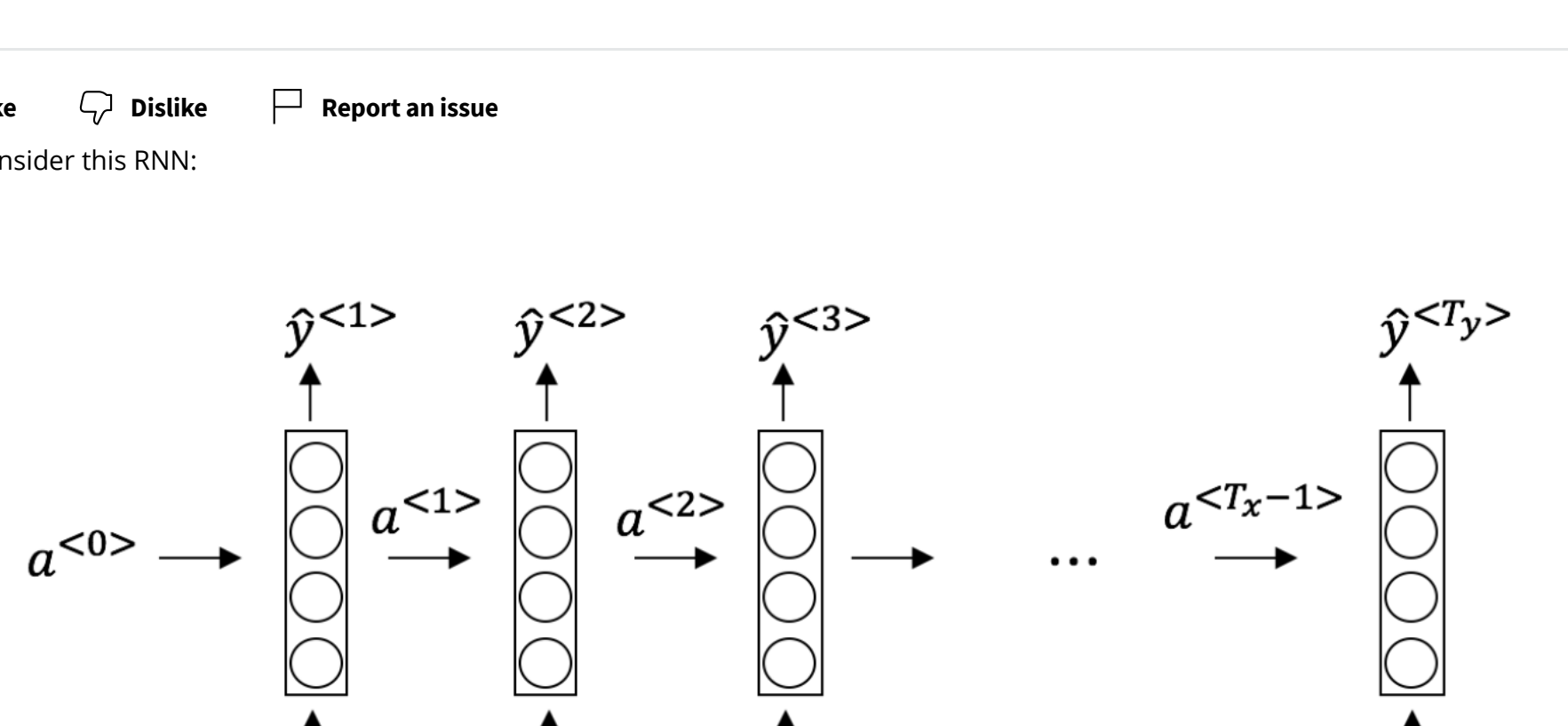
Like Dislike Report an issue

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 / 1 point**

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

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

2. Consider this RNN: **1 / 1 point**

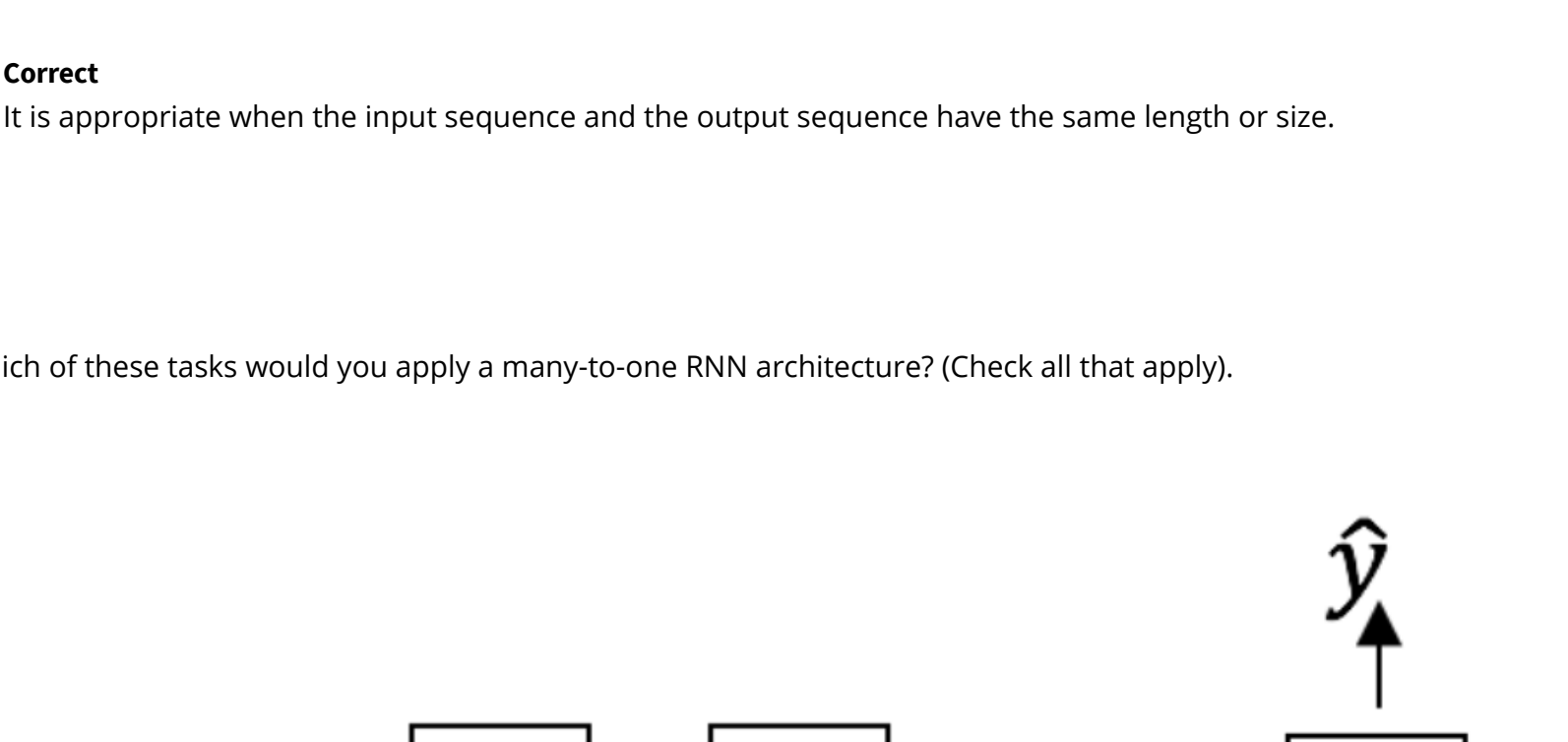


True/False: This specific type of architecture is appropriate when Tx=Ty

- ( ) False
- (●) True

Correct
It is appropriate when the input sequence and the output sequence have the same length or size.

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



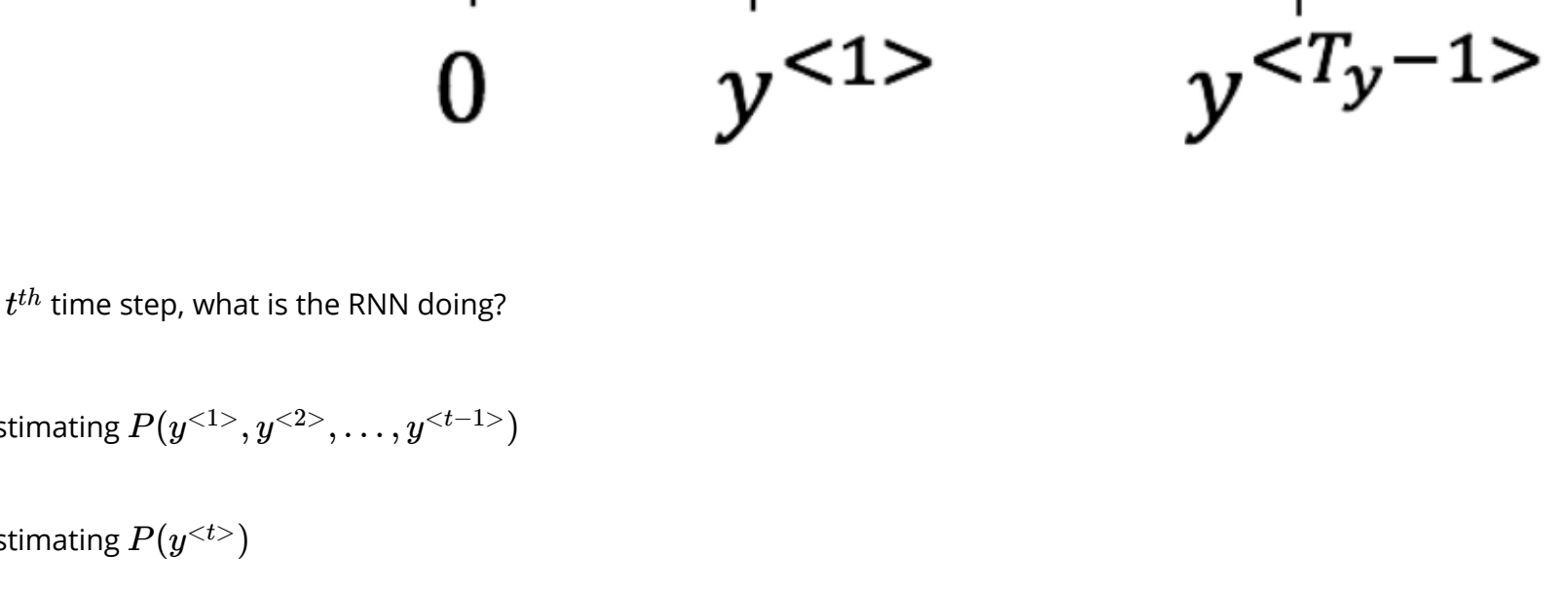
- [ ] Speech recognition (input an audio clip and output a transcript)
- [x] Sentiment classification (input a piece of text and output a 0/1 to denote positive or negative sentiment)

Correct
Correct!

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

Correct
Correct!

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

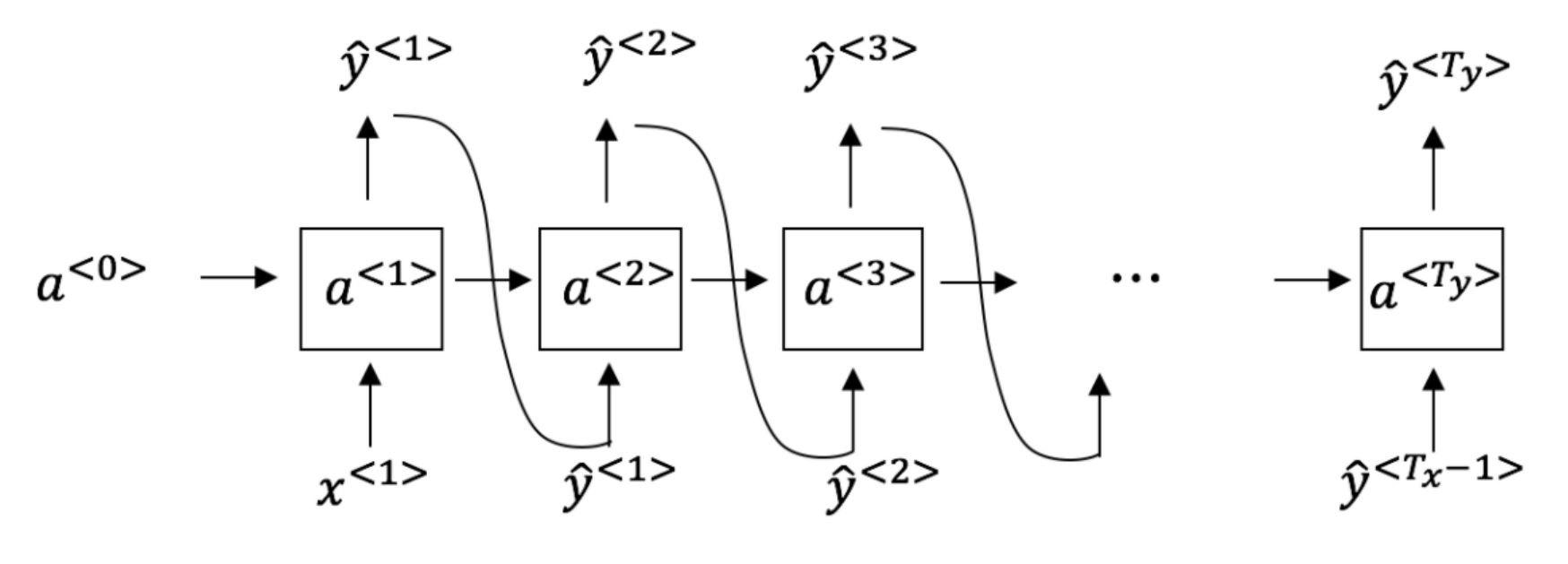


At the t^{th} time step, what is the RNN doing?

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

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 randomly sample a chosen word for that time-step. Then it passes this selected word to the next time-step.

- (●) True
- ( ) False

Correct
Step t uses the probabilities output by the RNN to randomly sample a chosen word for that time-step. Then it passes this selected word 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 a vanishing gradient problem. **1 / 1 point**

- ( ) True
- (●) False

Correct
Vanishing and exploding gradients are common problems in training RNNs, but in this case, your weights and activations taking on the value of NaN implies you have an exploding gradient problem.

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 Γ_u at each time step? **1 / 1 point**

- (●) 500
- ( ) 50000
- ( ) 200
- ( ) 5

Correct
Correct, Γ_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 always remove the Γ_u. I.e., setting $\Gamma_u = 0$. **1 / 1 point**

- (●) True
- ( ) False

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. 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. **0 / 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

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

- ( ) False
- (●) True

Incorrect
No. 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. 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→y*. Should you use a Unidirectional RNN or Bidirectional RNN for this problem? **1 / 1 point**

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

Correct