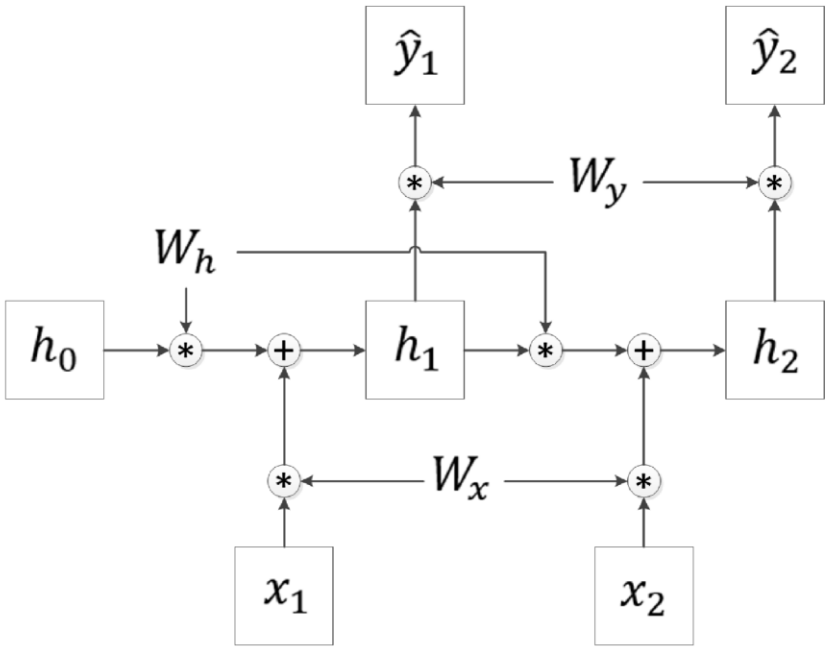
# CISC 616

# Assignment #3 (Due May 27, 2022 @ 11:59 PM)

1. (8 points) A RNN network is illustrated below (as a computational graph). Assume that the identity function is used to generate the output of all neurons.




Assume that the input at a given time, $h_0 = 1$, $x_1 = 10$, $x_2 = 10$, $y_1 = 5$, and $y_2 = 5$. The initial values of the weights are: $W_h = 1$, $W_x = 0.1$, $W_y = 2$. Answer the following questions:

1. Compute the predicted value $\hat{y}_2$.
2. Compute the total loss, $L_t = \sum_i (\hat{y}_i - y_i)^2$ for the given values of weights and inputs.
3. Compute the derivative $\frac{\partial L_t}{\partial h_1}$.
4. Compute the derivative $\frac{\partial L_t}{\partial W_h}$.

2. Why are long term dependencies difficult to learn in a RNN? You may use this equation to explain your answer.

$$h_t = \tanh(W_{hh} h_{t-1}, W_{xh} x_t)$$

3. (2 points) When would the use of Gated Recurrent Units (GRU) be more efficient than vanilla RNNs?

4. (2 points) What are the advantage and disadvantage of Truncated Backpropagation Through Time (TBTT)?

5. (8 points) You are required to define a simple RNN to decrypt a Caesar Cipher. A Caesar Cipher is a cipher that encodes sentences by replacing the letters by other letters shifted by a fixed size. For example, a Caesar Cipher with a left shift value of 3 will result in the following:

   Input: ABCDEFGHIJKLMNOPQRSTUVWXYZ
   Cipher: DEFGHIJKLMNOPQRSTUVWXYZABC

   Notice that there is a 1-to-1 mapping for every character, where every input letter maps to the letter below it. Because of this property you can use a character-level RNN for this cipher, although word-level RNNs may be more common in practice.
   Answer the following questions:

   a. A Caesar Cipher can be solved as a multiclass classification problem using a fully-connected feedforward neural network since each letter X maps to its cipher value Y. However, an RNN will perform much better. Why?

   b. Describe the nature of the input and output data of the proposed model.

   c. Will the model by a character-level one-to-one, one-to-many, or many-to-many (matching) architecture? Justify your answer.

   d. How should the training data look like? Give an example of a sample input and the corresponding output.

   e. What is a good way to handle variable length texts?

   f. In order for the model to function properly, the input text has to go through several steps. For example, the first step is to tokenize is the text, i.e., to convert it into a series of characters. What should be the other required steps in order to train the model?

   g. What should be the architecture of the simple RNN that can be used? You might use Keras API to describe the architecture.

   h. Are there any processing operations that should be applied to the output in order to generate the deciphered text?