Problem Set 3

The required weekly readings and lecture slides should be helpful in completing the assignment. You can find these on our course website. For your submission, you must submit a PDF that has been typed. For mathematical derivations, if easier, you can insert pictures of your handwritten derivation into the final PDF.

1. **Fine-Tuning [4 points]:**

   (a) Describe the motivation for using fine-tuning to develop the R-CNN object detection model and how fine-tuning was implemented.

   (b) Describe the motivation for using fine-tuning to develop the fully convolutional semantic segmentation model and how fine-tuning was implemented (before adding skip connections).

2. **Recurrent Neural Networks; i.e., RNNs [7 points]:**

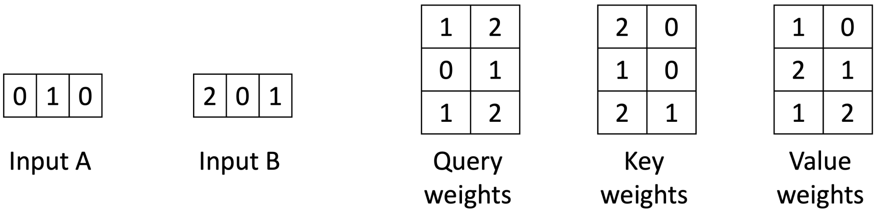
   (a) Identify one possible application, not covered in our lectures, for each of these type of sequence-based problems: one-to-many, many-to-one, and many-to-many.

   (b) Describe how the design of recurrent neural networks for each of the following sequence problems match and/or differ from each other: one-to-many, many-to-one, and many-to-many. Your response must indicate what should be the input and output of each network.

   (c) Assume you design a 3-layer RNN to predict a character sequence when given an input character sequence. What will happen to the number of model parameters when the number of input characters quadruples?

3. **Attention [12 points]:**

   - Compute all attention weights that result when using self-attention with the two inputs and weight matrices shown in Figure 1. Use the dot product to calculate the query-key similarity (Note: do not scale the dot product result, as often is done to arrive at more stable gradients during training). Mathematical steps must be shown for full credit.
   - Using your calculated attention weights from the previous step, compute the resulting representations of the input tokens from self-attention for both inputs using the weight matrices shown in Figure 1. Mathematical steps must be shown for full credit.



Input A

0	1	0

Input B

2	0	1

Query weights

1	2
0	1
1	2

Key weights

2	0
1	0
2	1

Value weights

1	0
2	1
1	2

Figure 1: Inputs and weight matrices for computing self-attention.

4. **Transformers [2 points]:** Describe two ways a transformer neural network is different from a recurrent neural network (RNN).

Collaboration versus Academic Misconduct: Collaboration with other students is permitted, but the work you submit must be your own. Copying/plagiarizing work from another student is not permitted and is considered academic misconduct. For more information about University of Colorado Boulder's Honor Code and academic misconduct, please visit the course syllabus.