Deep Learning / Spring 2024

Homework 2

Please upload your assignments on or before March 25, 2024.

- You are encouraged to discuss ideas with each other. But you **must acknowledge** your collaborator, and you **must compose your own** writeup and code independently.
- We **require** answers to theory questions to be written in LaTeX. (Figures can be hand-drawn, but any text or equations must be typeset.) Handwritten homework submissions will not be graded.
- We **require** answers to coding questions in the form of a Jupyter notebook. It is **important** to include brief, coherent explanations of both your code and your results to show us your understanding. Use the text block feature of Jupyter notebooks to include explanations.
- Upload both your theory and coding answers in the form of a **single PDF** on Gradescope.

---

1. **(3 points)** *Recurrences using RNNs.* Consider the recurrent network architecture below in Figure 1. All inputs are integers, hidden states are scalars, all biases are zero, and all weights are indicated by the numbers on the edges. The output unit performs binary classification. Assume that the input sequence is of **even** length. What is computed by the output unit at the final time step? Be precise in your answer. It may help to write out the recurrence clearly.

2. **(3 points)** *Understanding self-attention.* Let us assume the basic definition of self-attention (without any weight matrices), where all the queries, keys, and values are the data points themselves (i.e., $x_i = q_i = k_i = v_i$). We will see how self-attention lets the network select different parts of the data to be the "content" (value) and other parts to determine where to "pay attention" (queries and keys). Consider 4 orthogonal "base" vectors all of equal ℓ_2 norm a, b, c, d. (Suppose that their norm is β, which is some very, very large number.) Out of these base vectors, construct 3 tokens:

$$x_1 = d + b,$$
$$x_2 = a,$$
$$x_3 = c + b.$$

   a. (0.5 points) What are the norms of x_1, x_2, x_3?

   b. (2 points) Compute $(y_1, y_2, y_3) =$ Self-attention(x_1, x_2, x_3). Identify which tokens (or combinations of tokens) are approximated by the outputs y_1, y_2, y_3.

   c. (0.5 points) Using the above example, describe in a couple of sentences how self-attention that allows networks to approximately "copy" an input value

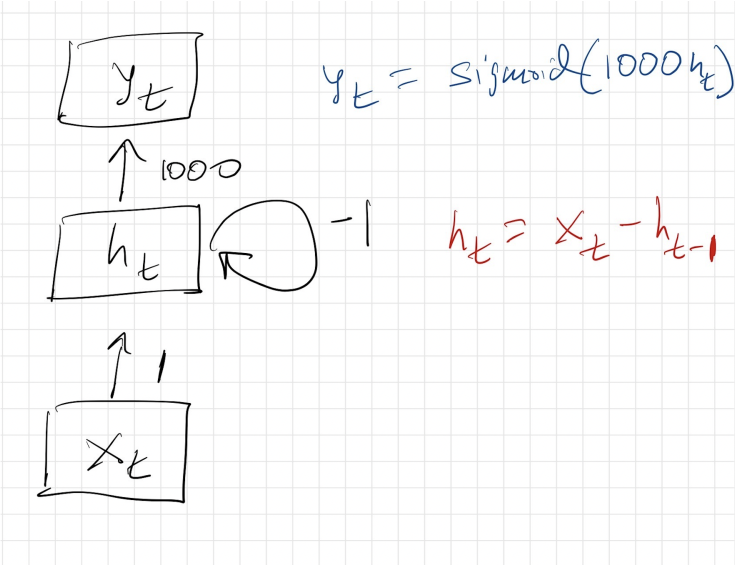

Figure 1: RNNs

to the output.

3. **(2 points)** *Attention! My code takes too long.* In class, we showed that a computing a regular self-attention layer takes $O(T^2)$ running time for an input with T tokens. One alternative is to use "linear self-attention". In the simplest form, this is identical to the standard dot-product self-attention discussed in the class and lecture notes, except that the exponentials in the rowwise-softmax operation softmax(QK) are dropped; we just pretend all dot-products are positive and normalize as usual. Argue that such this type of attention mechanism avoids the quadratic dependence on T and in fact can be computed in $O(T)$ time.

4. **(3 points)** *Vision Transformers.* In HW1 you trained a dense neural network which can classify images from the FashionMNIST dataset. In this problem, you are tasked to achieve the same objective, but using Vision Transformers. Use a patch size of 4x4, 6 ViT layers, and 4 heads. You can adapt the Jupyter notebook provided on Brightspace to train ViTs.

5. **(4 points)** *Sentiment analysis using Transformer models*. Open the (incomplete) Jupyter notebook provided as an attachment to this homework in Google Colab (or other cloud service of your choice) and complete the missing items. Save your finished notebook in PDF format and upload along with your answers to the above theory questions in a single PDF.