COMP4650/COMP6490 Document Analysis
2025 Semester 2

Computing Lab 4

Q1: Self-Attention

(This is a theory question that does not require coding.)

You are given a set of vectors below: $\mathbf{x}_1 = \begin{bmatrix} 2 \\ -1 \\ 3 \end{bmatrix}$, $\mathbf{x}_2 = \begin{bmatrix} 1 \\ 2 \\ 4 \end{bmatrix}$, $\mathbf{x}_3 = \begin{bmatrix} -2 \\ -3 \\ 0 \end{bmatrix}$, Compute the output sequence from a self-attention block using dot-product similarity $\text{sim}(\mathbf{x}, \mathbf{y}) = \mathbf{x}^\top \mathbf{y}$. (You do not need to project the vectors to separate key/query/value vectors. You can just use these $\mathbf{x}$ vectors for all three roles.)

Q2: Transformers

(This is a theory question that does not require coding.)

(a) What is the computational complexity (i.e., how many computation steps do you need to perform) for a self-attention block with vector dimension d and sequence length l?

(b) Is this complexity smaller or larger compared to a GRU with dimension d and sequence length l?

(c) Hence the transformer model can be trained in much shorter time, True or False?

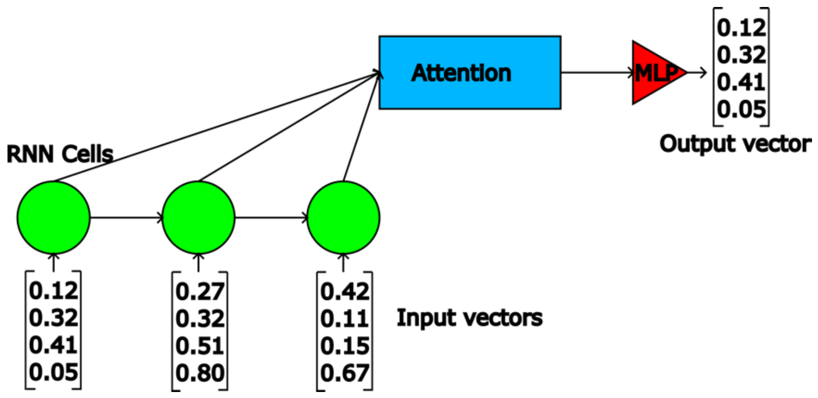
(d) Describe a type of task where a transformer model is likely to perform much better than a recurrent model.

(e) The transformer architecture repeats self-attention layers and then feed-forward layers. Explain the importance of each of these two types of layers.

Practical Exercise: Attention

In the provided notebook `lab4-memory_test_attention.ipynb` you will use attention with RNN models. Given a sequence of input vectors your network should output the first vector in the sequence. The models are run on a synthetic dataset, where inputs are sequences of random vectors and the label of a given sequence is the first vector of that sequence. In order to solve this task, the model must remember the first vector it sees and then ignore everything else.

The diagram below shows how a sequence of 3 input vectors is processed by the attention model (most tests use a longer sequence). Note that in the code an entire batch of examples is processed at once, thus the input to each cell is a matrix and the output is also a matrix.



Answer the following questions:

(a) Does the attention model perform better or worse than the models without attention?

(b) In lectures we saw many different types of attention score functions.

$$\text{score}(\mathbf{x}, \mathbf{y}) = \mathbf{x}^\top \mathbf{y}$$
$$\text{score}(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x}^\top \mathbf{y}}{\sqrt{d}}$$
$$\text{score}(\mathbf{x}, \mathbf{y}) = \mathbf{x}^\top W \mathbf{y}$$
$$\text{score}(\mathbf{x}, \mathbf{y}) = \mathbf{v}^\top \tanh(W\mathbf{x} + U\mathbf{y})$$

(i) Which one is being used here?

(ii) Try to implement at least two others? Do they work? Why/why not?

(c) Attention is typically normalised using the softmax function. Would it still work if instead you used the following function? Implement this ReLU normalisation $a_j = \frac{\text{ReLU}(\tilde{a}_j)}{\sum_{i=1}^{n} \text{ReLU}(\tilde{a}_i)}$ in the `AttentivePooling` and/or `AttentionGRU`. Does the new attention perform as you predicted?