seq2seq

Consider the following "summarization" problem. We have documents composed of tokens {"a", "b", "c", "d", "e"} and would like to train a function to generate the corresponding sequence of counts, i.e. the output is always of length 5 and consists of the number of "a"s, the number of "b"s, etc.

Examples

- "badcab." -> "22110."
- "bababacee." -> "33101."
- "dadda." -> "20030."

"." indicates a special end-of-sequence token.

Manually choose weights for a simple RNN encoder-decoder model to solve this problem.

$$\text{state}_0 = \mathbf{0}$$

$$\text{state}_{t+1} = \text{encode}(\text{token}_t, \text{state}_t)$$

$$\text{state}'_0 = \text{state}_T$$

$$\text{output}_v, \text{state}'_{v+1} = \text{decode}(\text{state}'_v)$$

If the token and hidden state are represented by column vectors x and h, respective:

$$\text{encode}(x, h) = W_e[x; h]$$

$$\text{decode}(h) = \text{ReLU}(W_o h), W_h h$$

Assume that the input tokens (including the EOS token) are one-hot encoded and the output is composed of a scalar count and a [0, 1] indicator of whether the sequence has ended. Identify what size you need for the inputs, outputs, and hidden state. Identify specifically what W_e, W_o, and W_h can be to solve this problem.

Note: you do not need to generate any data, write any code, or train any network. Your job is to *manually* identify weights that can solve this problem.

You may assume that there is a maximum sequence length of 100.