WEEKLY PARTICIPATION 12

In homework 5, you built a simple OCR system using a CNN + RNN architecture. The CNN outputs a sequence of vectors $\mathbf{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T]$; the RNN takes this sequence as input and ouputs a sequence $\mathbf{Z} = [\boldsymbol{z}_1, \ldots, \boldsymbol{z}_T]$. Each of these vectors are then converted by a linear layer into a vector of logits for the probabilities over the digits 0–9 and a special spacing token. This sequence of probabilities is then converted into a sequence of digits by using a CTC decoder.

(1) We used the CTC loss in homework 5 because we use the RNN hidden states as contextual representations, so there are T outputs in $\mathbf{Z}$ that need to be decoded into a sequence of digits of length possibly less than T.

Describe an alternative approach to get from $\mathbf{X}$ to the predicted digits, also using RNNs, that can be trained using the regular cross-entropy loss, so does not require the CTC loss.

(2) Now that you have learned about encoder transformers, you could replace the RNN with an encoder transformer over the CNN outputs to generate contextual embeddings $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_T$. Explain why it is a bad idea to replace $\mathbf{Z} = \texttt{RNN}(\mathbf{X})$ with $\mathbf{Z} = \texttt{Encoder}(\mathbf{X})$, and explain a simple modification that will work.