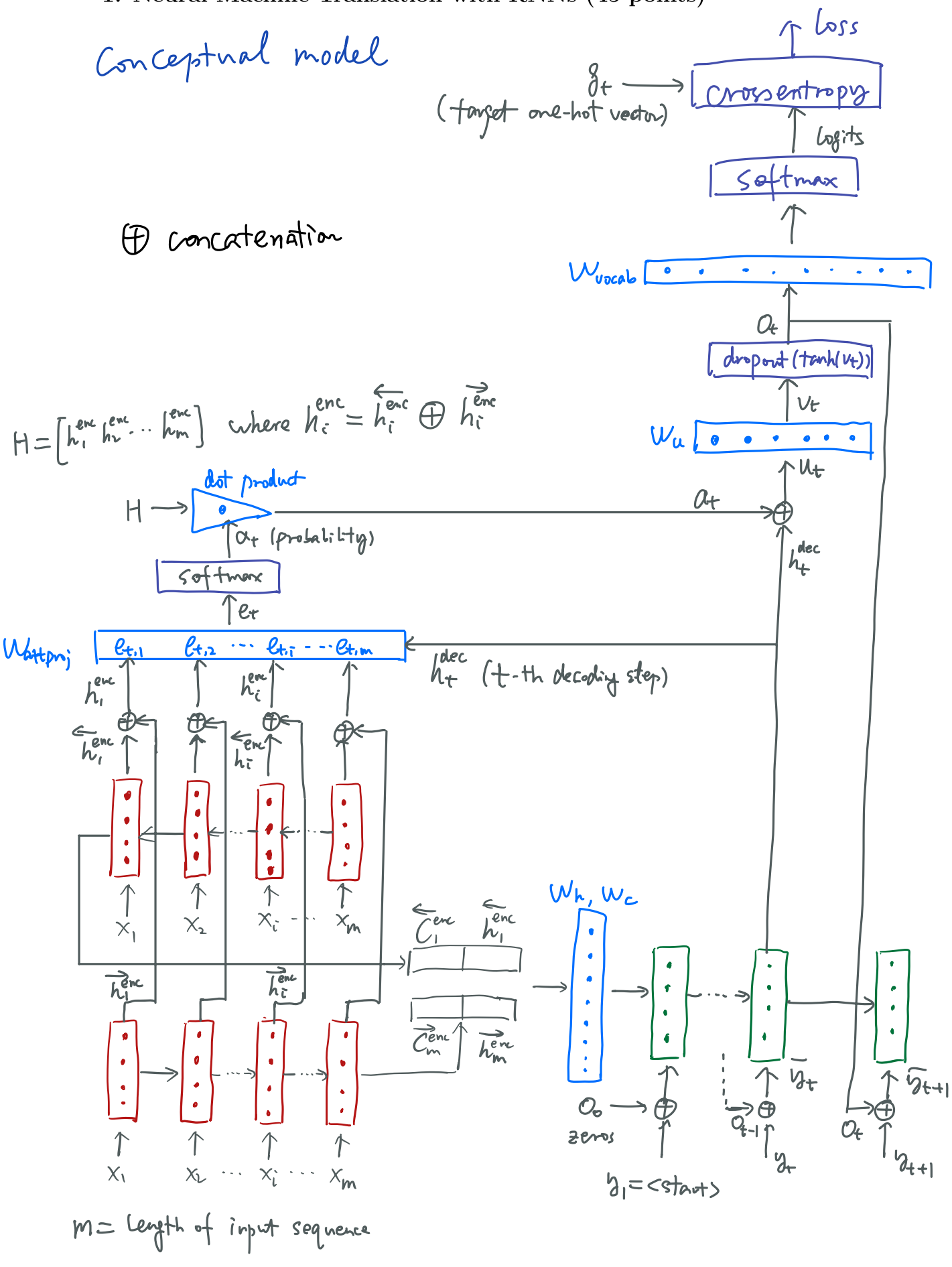
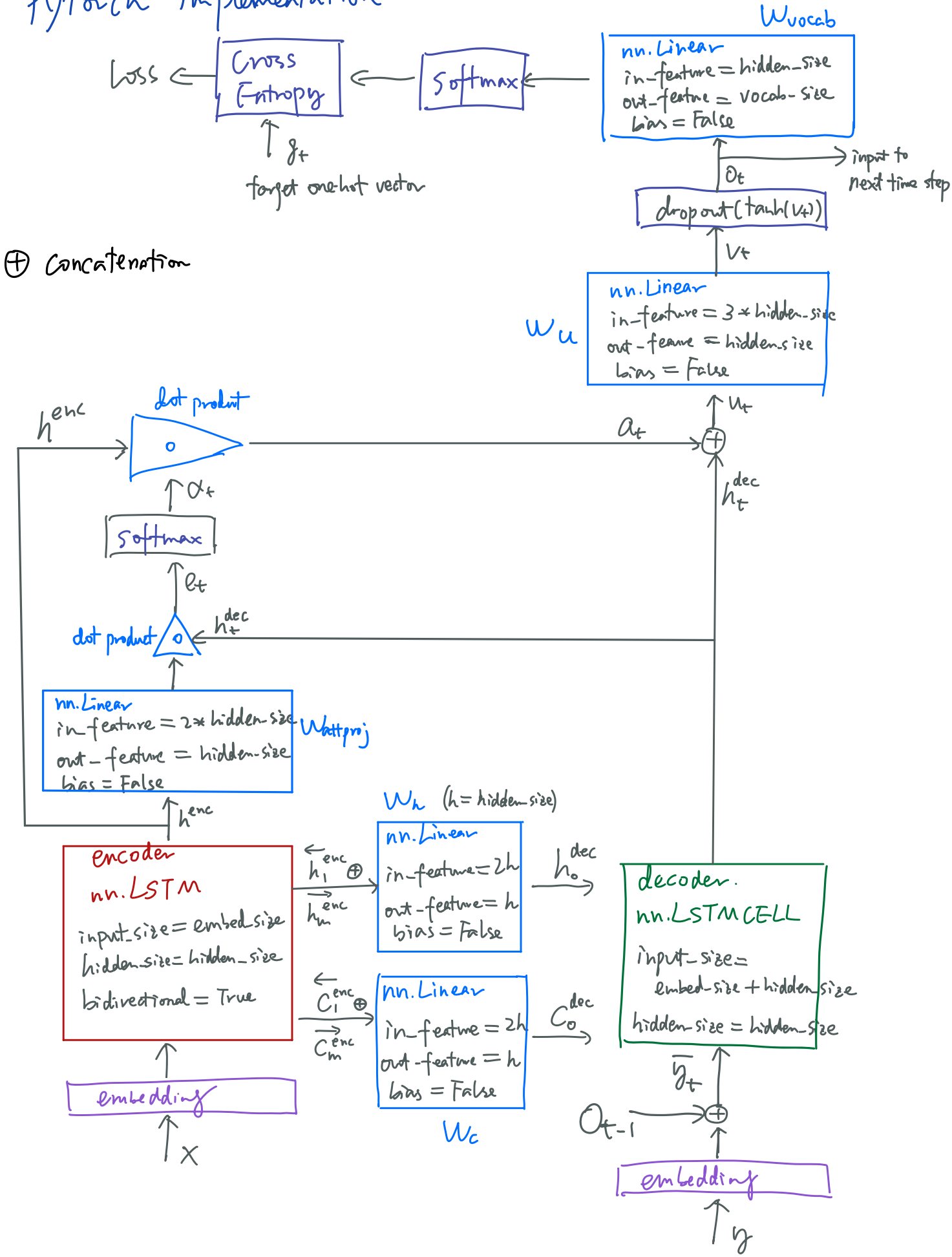
1. Neural Machine Translation with RNNs (45 points)



m = length of input sequence

Pytorch implementation



(g) (3 points) (written) The `generate_sent_masks()` function in `nmt_model.py` produces a tensor called `enc_masks`. It has shape (batch size, max source sentence length) and contains 1s in positions corresponding to 'pad' tokens in the input, and 0s for non-pad tokens. Look at how the masks are used during the attention computation in the `step()` function.

First explain (in around three sentences) what effect the masks have on the entire attention computation. Then explain (in one or two sentences) why it is necessary to use the masks in this way.

$e_{t,i}$'s which correspond to the 1s in enc_masks, are set to $-\inf$.
As a result, the corresponding $\alpha_{t,i}$'s become zero,
and the hidden states of the pad tokens become zero in attention a_t.
It is necessary because there should be no attention on the pad tokens and we don't want gradients for them too.

(i) (4 points) (written) In class, we learned about dot product attention, multiplicative attention, and additive attention. As a reminder, dot product attention is $\mathbf{e}_{t,i} = \mathbf{s}_t^T \mathbf{h}_i$, multiplicative attention is $\mathbf{e}_{t,i} = \mathbf{s}_t^T \mathbf{W} \mathbf{h}_i$, and additive attention is $\mathbf{e}_{t,i} = \mathbf{v}^T \tanh(\mathbf{W}_1 \mathbf{h}_i + \mathbf{W}_2 \mathbf{s}_t)$.

i. (2 points) Explain one advantage and one disadvantage of *dot product attention* compared to multiplicative attention.

ii. (2 points) Explain one advantage and one disadvantage of *additive attention* compared to multiplicative attention.

i. Multiplicative attention does a linear transformation on h_i, the encoder hidden states, and then takes the dot product with s_t, the decoder hidden states. It should work better in most cases than taking the dot product directly between s_t and h_i, since s_t and h_i can be in different spaces. But since Multiplicative attention introduces a linear layer W, it needs more data to train.

ii. Additive attention seems to step further than Multicative attention. My intuition is that it may work better than multiplicative attention in more complex problems where decoder and encoder states are very different and a linear transformation can not transform encoder states into a space that decoder states live.