CS224N Assignment 3 - Written Solutions

Question (g): Attention Masks

Effect of Masks on Attention Computation

The masks have a critical effect on the attention mechanism by preventing the model from attending to padding tokens in the source sentence. Specifically, when the attention scores e_t are computed, the mask sets all scores corresponding to padding positions to negative infinity ($-\infty$) before the softmax operation is applied. When softmax is subsequently applied to these scores (line 384), the $\exp(-\infty) = 0$ property ensures that padding positions receive zero attention weight in α_t, meaning they contribute nothing to the attention context vector a_t that is computed as a weighted sum of encoder hidden states.

Why Masks Are Necessary

Masking is necessary because padding tokens are artificial placeholders added to make all sentences in a batch the same length, and they contain no meaningful linguistic information. If the model were allowed to attend to these padding positions, it would introduce noise into the attention mechanism and potentially learn spurious patterns, degrading the model's ability to focus on the actual content of the source sentence and ultimately hurting translation quality.

Question (h): Model BLEU Score

After training the model and running `sh run.sh test`, the corpus BLEU score is: **[INSERT YOUR BLEU SCORE HERE]**

Note: The BLEU score should be larger than 18. Run the test command and fill in the actual value.

Question (i): Attention Mechanisms Comparison

Part (i): Dot Product vs Multiplicative Attention

Advantage of Dot Product Attention: Dot product attention is computationally simpler and more efficient because it requires no learnable parameters. It directly computes the similarity between s_t and h_i through $e_{t,i} = s_t^T h_i$, making it faster to compute and requiring less memory than multiplicative attention which needs to learn and store the weight matrix W.

Disadvantage of Dot Product Attention: Dot product attention is less flexible and expressive because it assumes that the decoder state s_t and encoder hidden state h_i are already in compatible semantic spaces for direct comparison. In contrast, multiplicative attention with $e_{t,i} = s_t^T W h_i$ can learn a transformation matrix W that projects the hidden states into a shared

space optimized for computing relevance, allowing it to better capture complex relationships between the decoder and encoder representations.

Part (ii): Additive vs Multiplicative Attention

Advantage of Additive Attention: Additive attention is more expressive and can model non-linear interactions between the decoder state and encoder hidden states through the use of the tanh activation function. The formulation $e_{t,i} = v^T \tanh(W_1 h_i + W_2 s_t)$ allows it to learn complex, non-linear scoring functions and can handle cases where the dimensions of h_i and s_t are different (since W_1 and W_2 can have different input dimensions).

Disadvantage of Additive Attention: Additive attention is computationally more expensive and has significantly more parameters to learn compared to multiplicative attention. It requires three parameter matrices (W_1, W_2, and v) instead of just one (W), leading to increased memory usage, longer training time, and a higher risk of overfitting, especially with limited training data.