True/False

1. T F To efficiently train a transformer decoder, we use teacher forcing and an attention mask over the output encodings after the current output, to keep the network "causal". True

2. T F Because self-attention parallelizes across the sequence while an RNN must be unrolled sequentially, a transformer generally achieves higher training throughput per epoch on parallel hardware. True

3. T F While training a machine translation task using Transformers, it is important that you prevent the decoder state from attending to future positions by masking. True

4. T F The advantage of LSTMs over Transformers is that you can feed words in parallel into the LSTM. False

5. T F A forget gate output of exactly 0 for a given cell-state dimension means that dimension's memory is preserved unchanged into the next timestep. False

6. T F The input gate i_t alone determines the new content written into the cell state. False

7. T F The output gate modifies the contents of the cell state C_t before it is passed to the next timestep. False

8. T F An RNN applies the same weight matrices at every timestep, regardless of sequence length. True

9. T F Because an RNN processes longer sequences, a longer input sequence requires the model to have more parameters. False

10. T F The vanishing-gradient problem in vanilla RNNs arises because backpropagation through time repeatedly multiplies by the recurrent weight matrix (and the activation Jacobian), so gradients shrink or grow exponentially in the number of timesteps. True

11. T F Vanishing and exploding gradients arise from the same underlying failure and are addressed by the same fix. False

12. T F The forward pass of an RNN cannot be parallelized across the time dimension the way a Transformer's self-attention can. True

Multiple Choice

1. Which of the following is not true of a Transformer encoder network?

(a) It is sequential in nature like a Recurrent Neural Network.
(b) Its operation can be parallelized.
(c) Position-embedding in Transformer encoders is required to introduce order.

Answer: A

2. Which of the following is not true for Transformers?

(a) The encoder can process the data in parallel during training.
(b) The decoder can generate the sequence in parallel during testing.
(c) Positional encoding helps in encoding the positional information of the input tokens.
(d) They easily overfit on a small dataset.

Answer: B

3. Which of the following is true for attention mechanisms?

(a) They use a Key to match a Value.
(b) In Transformers, all of the matrices for one attention head are shared across positions.
(c) They do not include a softmax layer.
(d) By default, they keep track of the relative position of inputs.

Answer: B

4. Which of the following is true for attention mechanisms?

(a) They use a Key to match a Value.
(b) In Transformers, all of the matrices for one attention head are shared across positions.
(c) They do not include a softmax layer.
(d) By default, they keep track of the relative position of inputs.

Answer: B

5. Transformers add positional encodings to the input embeddings primarily because:

(a) The embeddings would otherwise be too large to fit in memory.
(b) They prevent the decoder from attending to future positions.
(c) Self-attention would otherwise have no notion of token order (it is permutation-equivariant).
(d) They normalize the input to zero mean and unit variance.

Answer: C

6. Transformers add positional encodings to the input embeddings primarily because:

(a) Self-attention is does not keep track of the order of inputs, and otherwise has no notion of token order.

(b) The embeddings would otherwise be too large to fit in memory.

(c) They prevent the decoder from attending to future positions.

(d) They normalize the input to zero mean and unit variance.

Answer: A

7. For a sequence of length n and model dimension d, the time/memory cost of a single self-attention layer scales as:

(a) $\mathcal{O}(n \cdot d^2)$

(b) $\mathcal{O}(n^2 \cdot d)$

(c) $\mathcal{O}(n \cdot d)$

(d) $\mathcal{O}(n^2 \cdot d^2)$

Answer: B

8. The primary motivation for using multiple attention heads rather than one is to:

(a) Increase the total parameter count for greater capacity

(b) Allow each head to process a different token in the sequence

(c) Parallelize attention across multiple GPUs

(d) Let the model jointly attend to information from different representation subspaces / relationship types

Answer: D

Acknowledgments

Several questions in this practice exam were adapted from Prof. Gary Cottrell.