Natural Language Understanding, Generation, and Machine Translation (2024–25)

School of Informatics, University of Edinburgh
Alexandra Birch and Shay Cohen

Tutorial 2: Transformers (Week 6)

Questions on this worksheet that ask for an expression or calculation generally have one correct answer, and are designed to concretely explore aspect of these models just beyond what we discussed in lecture, by asking you about some implications of model design. These questions are not intended to be difficult, but notice that they are not primarily about recalling information—they require you to engage with the material and pay attention to the details. You should expect that some of the *easier* exam questions may be of this form.

We also include some more open-ended questions that you will need to think about. It is intended to help you see how modelling choices impact the number of parameters and how this impacts your design decisions when applying the Transformer model to a task. Most of these questions are answerable from combined understanding of Lectures Lectures 7 & 8 as well as your experience of Coursework 1. For a complete understanding, we advise reviewing Attention is all you Need, the paper which proposed the Transformer model, before beginning this tutorial.

1 Transformer's Efficiency

Question 1:
We now want to compare theoretical complexities between different models used in NLP tasks. Inspect Table 1 in Attention is all you Need with a focus on the "Complexity per Layer" column wherein n is the sequence length and d is the representation dimension, the same parameter as the self-attention projection size from Q1.

a. Consider the complexity bounds and your own knowledge of how NLP tasks are constructed – describe when a Transformer network might have lower complexity than other networks.

b. Other than complexity – describe other constraining factors to be considered when planning experiments using neural networks for NLP. There is no right answer here, state your own ideas.

Question 2:
One of the greatest advantages of the Transformer model is that it is possible to parallelise its computation, and therefore train a model on much larger training datasets (for example, for pre-training) with less compute.

a. Consider an encoder-decoder RNN model, following up on the question before. Can the computation of the encoder be easily parallelised with respect to the number of tokens (meaning, can you break the input and the computation into chunks such that they run in parallel)? Explain why or why not.

b. Consider the Transformer encoder layer. Explain why its computation can be parallelised over tokens *per layer*. Can computation be easily parallelised across layers?

2 Considering Permutations

The Transformer self-attention routine operates on *sets* of inputs without respect for sequence ordering. This model will produce identical outputs with varying combinations of inputs because the attention states between any two words will be the same regardless of word position. This gives the Transformer some interesting mathematical properties we want you to think about here.

Question 3:

a. For a transformer encoder model without positional embeddings – explain why the model is permutation **equivariant** but not **invariant**.

Consider this model with an additional max pooling layer on the output to combine the outputs to produce one output vector.

c. Explain why the whole model is now permutation **invariant** and not **equivariant**.

These properties are not desirable for sequence modelling in natural language processing. For this reason, we augment the inputs with **positional embeddings** to provide additional information to the input.

d. What additional information is provided with this addition and how does it help sequence modelling? Explain your answer in relation to how the properties described above are affected.