Congratulations! You passed!

Grade received 100% **To pass** 80% or higher

Go to next item

1. Which of the following are issues with transformers? **1 / 1 point**

- [ ] They allow for parallel computing.
- [x] Attention on sequence of length L takes L^2 time and memory.

Correct
Correct.

- [x] N layers take N times as much memory.

Correct
Correct.

- [ ] They help with the vanishing gradient problem.

2. Why do we need to store activations somewhere when implementing the transformer network? **1 / 1 point**

- ( ) We will need to keep track of all the activations we used so we can make predictions.
- (•) We need to save them to compute the back-propagation.
- ( ) They are important for interpretability.
- ( ) To allow us to debug our model incase it stops working.

Correct
Correct.

3. Why do we use locality sensitive hashing when computing attention? **1 / 1 point**

- [x] It allows us to not have to compare each query with each key. Instead we only compare the vectors that are found in the same bucket.

Correct
Correct.

- [x] It is a faster way to compute attention.

Correct
Correct.

- [ ] It is more accurate when finding the most similar vectors than regular attention.
- [ ] It is not worth using.

4. What is the point of using reversible layers? **1 / 1 point**

- ( ) It allows you to have a symmetry in your model, and thus breaks it in the backprop.
- ( ) It allows your model to capture dependencies that you would not have been able to capture otherwise.
- (•) It allows you to reconstruct the the activations and as a result you do not have to save them.
- ( ) It speeds up training.

Correct
Correct.

5. Standard Transformer is defined as: **1 / 1 point**

$$y_a = x + Attention(x)$$

$$y_b = y_a + FF(y_a)$$

Reversible:

$$y_1 = x_1 + Attention(x_2)$$

$$y_2 = x_2 + FF(y_1)$$

To recompute x_1 from y_1 you can use the following:

$$x_1 = y_1 - Attention(x_2)$$

How would you recompute x_2?

- (•) $x_2 = y_2 - FF(y_1)$
- ( ) $x_2 = Attention(x_1) + FF(y_1)$
- ( ) $x_2 = x_1 - FF(y_2)$
- ( ) $x_2 = y_2 - Attention(x_1)$

Correct
Correct.

6. Select two main components that the reformer uses which makes it more efficient than the transformers. **1 / 1 point**

- [x] Reversible layers

Correct

Correct.

- [x] Locality sensitive hashing.

Correct
Correct.

- [ ] K-nearest neighbors
- [ ] Skip connections.

7. What are the pros and cons of having more hashes when implementing LSH? **1 / 1 point**

- ( ) The more hashes you have the less accurate your model is, but the faster it is.
- (•) The more hashes you have the more accurate your model is, but the slower it is.
- ( ) The more hashes you have the faster you can train your model, and the more accurate it gets.
- ( ) The more hashes you have the slower your model gets and the lower the accuracy becomes.

Correct
Correct.

8. How many words can a reformer hold on a single 16GB GPU? **1 / 1 point**

- ( ) 500,000
- ( ) 200,000
- (•) 1 million
- ( ) 50,000

Correct
Correct.

9. In LSH, you want to attend to a bucket in a previous chunk because it covers the case with a hash bucket that is split over more than 1 chunk. **1 / 1 point**

- ( ) False.
- (•) True.

Correct
Correct.

10. One reason, according to the lecture why the BLEU score for Reversible Transformers is slightly better than the original Transformers, is due to parameter tuning in the 3 years since the original Transformer paper was published. **1 / 1 point**

- (•) True.
- ( ) False

Correct
Correct.