ARCHITECTING INTELLIGENCE ASSN 2:

Q1) Why can't a perceptron solve XOR? What changed with MLPs?

A perceptron can only draw one straight decision boundary to separate data.
The XOR problem needs a non-linear separation, which means no single straight line can classify all the points correctly. When multi-layer perceptrons were introduced, hidden layers with non-linear activations were added. These layers allow the network to break the problem into parts and learn more complex shapes, which is why XOR becomes solvable.

Q2) Why do stacked linear layers stay linear? Why do gradients vanish? Why does ReLU help?

If you stack multiple linear layers without any activation in between, the overall operation is still just one linear transformation. So stacking linear layers alone doesn't increase the model's power. Gradients shrink in deep networks mainly because activations like sigmoid or tanh squash values into small ranges. During backpropagation, their small derivatives keep getting multiplied, causing gradients to become extremely small as they move backward. ReLU works better because its gradient is not small for positive values, so information flows backward more easily and deep networks can be trained more effectively.

Q3) Why is positional encoding needed? Sinusoidal vs absolute PE? Why does RoPE help?

Transformers process all tokens at once, so by default they don't know the order of words in a sentence. Positional encoding is added to give the model information about where each token appears. Absolute positional encoding assigns a fixed or learned vector to each position, while sinusoidal encoding uses sine and cosine functions so the model can generalize to longer sequences. RoPE rotates the query and key vectors based on position, which naturally captures relative distances between tokens. This makes it especially useful for handling longer contexts.

Q4) What are Query, Key and Value? Why scale by √dk? Why are diagonal values high?

Query represents what a token is looking for, Key represents what a token offers, and Value is the actual information that gets passed forward. The attention score is divided by √dk to prevent the dot products from becoming too large, which would make the softmax overly confident and hurt learning. Diagonal values are usually high because each token tends to attend most strongly to itself, especially in earlier layers.

Q5) Why split attention into multiple heads? What do d_model, h and d_head mean?

Using multiple attention heads allows the model to focus on different types of relationships at the same time, such as grammar, meaning, or long-range dependencies.

- **d_model** is the total embedding size
- **h** is the number of attention heads
- **d_head** is the dimension handled by each head, calculated as d_model / h

Q6) Why is greedy decoding suboptimal? Why is beam search better? Give an example.

Greedy decoding chooses the most probable word at each step, without considering how it affects the rest of the sentence. This can lead to locally good but globally poor results. Beam search keeps multiple possible sequences at each step and selects the best overall sequence at the end, which usually produces more meaningful sentences.

For example, greedy decoding might end a sentence too early, while beam search explores longer options and produces a more complete and fluent result.