Classwork: Transformer Explainer

⚠ This is a preview of the published version of the quiz

Started: Nov 20 at 11:49pm

Quiz Instructions

Transformer Explainer

Objectives:

Understand how and why LLMs work

Purpose:

In this assignment, you will explore and understand the Transformer model architecture through the Transformer Explainer tool. By completing this assignment, you will gain a deeper understanding of how Transformer models process text and the key components involved in their operation.

Instructions:

1. **Visit the Transformer Explainer:**
   - Go to the Transformer Explainer tool: **Transformer Explainer (https://poloclub.github.io/transformer-explainer/)** .
   - Spend time interacting with the tool and familiarize yourself with the different components of the Transformer model.
2. **Complete the Reading:**
   - Read through the content provided in the Transformer Explainer tool to understand how each part of the model works, including tokenization, embeddings, self-attention mechanisms, and the final output generation process.
3. **Review the Key Components:**
   - **Embedding**: Understand how the input text is tokenized and converted into embeddings.
   - **Transformer Block**: Learn about the core building blocks like self-attention and multi-layer perceptron (MLP) layers.
   - **Attention Mechanism**: Focus on how self-attention allows the model to prioritize important

tokens.
    - **Output Probabilities**: See how the final output is generated based on the processed information.

4. **Answer the Questions Below:** After completing the reading and interacting with the tool, answer the following questions based on your understanding.

Question 1 2 pts

GPT-2 has a vocabulary of _____ unique tokens.

Question 2 3 pts

Match the following components with their descriptions:

Tokenization [Choose]

Self-Attention [Choose]

Multi Layer Perceptron [Choose]

Question 3 2 pts

What is the primary function of the attention mechanism in the Transformer model?

To process the text in a sequential manner

To output probabilities for the next token

To focus on the most relevant parts of the input

To tokenize the input text

Question 4 1 pts

The **Transformer** model was first introduced in the paper "_____ is All You Need" in 2017.

Question 5 1 pts

What is the main principle behind text-generative Transformer models?

- Image generation
- Sound recognition
- Next-sentence prediction
- Next-word prediction

Question 6 1 pts

What is the core innovation of the Transformer model?

- Convolutional Neural Networks (CNNs)
- Recurrent Neural Networks (RNNs)
- Backpropagation
- Self-attention mechanism

Question 7 1 pts

What is the main purpose of the Multi-Layer Perceptron (MLP) layer in a Transformer model?

- Capture relationships between tokens
- Embed tokens
- Generate text
- Process tokens independently

Question 8 2 pts

What happens when the temperature in the Transformer model is greater than 1?

- The model becomes less confident and more random
- The model generates lower probabilities for all tokens
- The model reduces the number of possible outputs
- The model becomes more confident and deterministic

Question 9 3 pts

Match the components of a Transformer architecture with their responsibilities.

Embedding	[Choose]
Transformer Block	[Choose]
Output Probabilities	[Choose]

Question 10 1 pts

What does the "softmax" function do in the Transformer model?

- It splits the input text into tokens
- It calculates attention scores
- It converts logits into probabilities
- It generates token embeddings

Question 11 2 pts

Which of these is NOT part of a Transformer block?

- Query-Value matrices
- Multi-Head Self-Attention

Multi-Layer Perceptron

Embedding layer

Question 12 3 pts

Match the following terms with their meanings:

Query [Choose]

Key [Choose]

Value [Choose]

Question 13 1 pts

What does the final linear layer in the Transformer architecture do?

It performs tokenization

It transforms the processed embeddings into probabilities

It generates the input embeddings

Question 14 2 pts

The temperature hyperparameter controls the sharpness of the probability distribution generated by the model.

True

False