CODE BASICS

DEEP LEARNING TAKEAWAYS

codebasics.io

Chapter:

Transformers

Word Embeddings

1. Word embedding is a way to represent a word in a numeric format such that it captures the semantic meaning of that word
2. Word2vec, Glove etc are popular techniques to produce static word embeddings
3. With word embeddings, you can do math like King - man + woman = Queen or Russia - Moscow + Delhi = India

Overview of Encoder and Decoder

1. Transformer architecture has two parts (1) Encoder (2) Decoder
2. The purpose of encoder is to produce a contextual embeddings for each word (more precisely a token) in a given input sentence
3. The purpose of decoder is to produce an output sequence which can be a word (for the next word prediction task) or a sequence (such as translated sentence in case of language translation)
4. BERT and GPT are examples of specific models that are based on a transformer architecture

Attention Mechanism

1. Attention mechanisms allow Transformers to focus on relevant parts of the input sequence for each output, improving context understanding.
2. Self-attention computes the relationships between all input elements, capturing dependencies regardless of their position.
3. Key components of attention include queries, keys, and values, which determine how much focus is given to different parts of the input.

Multi Headed Attention

1. Multi-headed attention enables Transformers to capture diverse relationships in the data by learning multiple attention patterns simultaneously.
2. Each attention head computes self-attention independently, focusing on different parts of the input sequence.
3. Outputs from all heads are concatenated and transformed to create a richer representation of the input.
4. Multi-headed attention improves the model's ability to understand complex patterns and long-range dependencies.
5. It is a key component in Transformer Architecture.

Decoder

1. The decoder in Transformer architecture generates the output sequence step-by-step, one token at a time.
2. It uses masked self-attention to ensure predictions depend only on previously generated tokens.
3. The decoder integrates encoder outputs through cross-attention to incorporate contextual information from the input sequence.
4. Fully connected layers in the decoder refine the processed information for final token prediction.
5. The decoder is central to tasks like language translation and text generation, where sequential output is crucial.

How Transformers are Trained?

1. In Self-supervised learning, labels are generated from the data itself without requiring manual annotations.
2. Casual Language Modeling (CLM) and Mask Language Modeling (MLM) are self-supervised learning approaches used to train transformers
3. GPT uses CLM whereas BERT uses MLM