Understanding Transformers: A Comprehensive Guide to Sequence Transduction with Self-Attention

Drawbacks of LSTMs and RNNs: (wrt Sequence Transduction)

1. LSTMs and RNNs present 3 problems:

    - Sequential computation inhibits parallelization
    - No explicit modeling of long and short range dependencies
    - “Distance” between positions is linear

2. RNN using Attention:

    - processing inputs (words) in parallel is not possible. For a large corpus of text, this increases the time spent translating the text.

3. Convolutional Neural Networks:

    Convolutional Neural Networks help solve these problems. With them we can

    - Trivial to parallelize (per layer)
    - Exploits local dependencies
    - Distance between positions is logarithmic

    Some of the most popular neural networks for sequence transduction, Wavenet and Bytenet, are Convolutional Neural Networks.

    The problem is that Convolutional Neural Networks do not necessarily help with the problem of figuring out the problem of dependencies when translating sentences. That’s why Transformers were created, they are a combination of both CNNs with attention.

Finally,

Transformers:
To solve the problem of parallelization, Transformers try to solve the problem by using encoders and decoders together with attention models. Attention boosts the speed of how fast the model can translate from one sequence to another.

Key Elements of a Transformer:

Self-Attention:

- Self-attention mechanism allows the model to weigh different words in the input sequence when making predictions.
- It captures relationships between words by assigning attention weights to each word based on its relevance to other words in the sequence.

Encoder-Decoder Structure:

- Transformers consist of an encoder and a decoder.
- The encoder processes the input sequence and generates a hidden representation, capturing the input's meaning.
- The decoder takes the encoder's hidden representation and generates the output sequence.

Positional Encoding:

- Positional encodings are added to the input sequence to provide information about the relative positions of words.
- They help the model distinguish between words based on their positions in the sequence.
- Positional encodings are usually fixed and added as embeddings to the input.

Multi-Head Attention:

- Multi-head attention is a variant of self-attention that allows the model to focus on different word dependencies simultaneously.
- It applies self-attention multiple times in parallel, each with a different set of learned weights, capturing different types of relationships.

Feed-Forward Neural Networks:

- Transformers include feed-forward neural networks as a component in both the encoder and decoder.
- These networks process the hidden representations and provide non-linear transformations, enabling the model to learn complex patterns.

Layer Normalization:

- Layer normalization is applied after each sub-layer (self-attention and feed-forward layers) to normalize the hidden representations.
- It helps stabilize training and improves the model's ability to generalize.

Residual Connections:

- Residual connections are added around each sub-layer, allowing the model to retain information from earlier layers.
- They help mitigate the vanishing gradient problem and improve information flow through the network.

Position-wise Feed-Forward Networks:

- Position-wise feed-forward networks are applied to each position in the encoder and decoder layers independently.
- They provide additional non-linear transformations to each word's hidden representation.

These are some of the key elements and concepts that form the foundation of a Transformer architecture. They enable the model to capture dependencies, parallelize computations, and improve the quality of sequence transduction tasks.