Deep Learning

Transformer Models for NLP

Learning Objectives

By the end of this lesson, you will be able to:

- Illustrate the concept of transformer models
- Examine the architectures of transformer models
- Design architecture and use cases of the BERT model
- Classify text using BERT

Business Scenario

A travel booking company has been struggling with the quality of its search engine results. Many of the user queries were returning irrelevant or incomplete results.

To solve this problem, it turned to the power of natural language processing (NLP) and decided to implement BERT, a transformer-based language model that has been proven to perform well on a variety of NLP tasks.

To train the BERT model, the company used a Masked Language Modeling approach where 15% of the words in a sentence were replaced with a MASK token to teach the model to predict missing words. With the new and improved search engine, customers are now getting more accurate and personalized search results, resulting in increased customer satisfaction and sales.

Overview of Transformer Models

Discussion

Discussion: Transformer Models

Duration: 10 minutes

- What are transformer models?
- What are some applications of transformer learning?

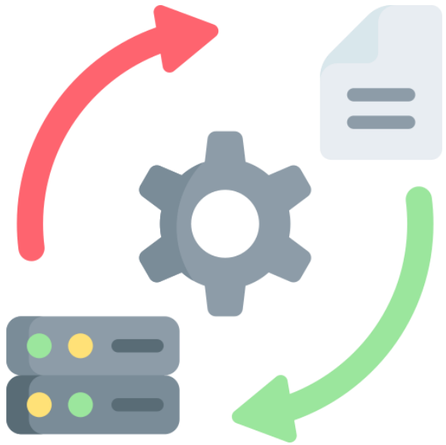
Transformer Models

Transformer models use self-attention mechanisms for processing sequential data, significantly enhancing natural language processing tasks.

An example is OpenAI’s GPT-3, which leverages the Transformer architecture to excel in tasks such as text completion, translation, and question answering.

Transformer Models

Long-range memory dependencies are a notable characteristic of transformers.



This functionality is acquired through the concept of self-attention.

Prior models such as RNNs and LSTMs had challenges dealing with long-range dependencies in sentences, which transformer models resolved.

Long-Range Dependencies

Consider the following sentence:

"Frenny likes to play basketball, he is really good at dunking."

Frenny is indeed the subject

he is a pronoun referring back to Frenny

RNNs and LSTMs are generally proficient at establishing such associations, especially when dealing with shorter sentences.

Long-Range Dependencies

Consider a bigger paragraph:

"Lara is a cook at McTown French Fries. She's been working there for three years now. The place immediately gained fame once she joined. Nobody knew of its existence until she joined there. She made good friends with other cooks and learned to cook various new dishes, which customers really enjoyed at that place."

- Lara
- McTown French Fries
- Pronouns of Lara
- Pronouns of McTown French Fries

Longer sequences or paragraphs can be challenging for traditional sequence models like RNNs and LSTMs to process.

Long-Range Dependencies

The algorithms (RNNs and LSTMs) often struggle to determine the subject of a sentence when processing a whole paragraph.

Eventually, it loses its association with other subjects in the input.

Transformer models retain the context of pronoun references and exhibit strong proficiency in recognizing the grammar of provided sentences.

Long-Range Dependencies

Uses:

- Natural Language Processing
- Computer Vision

Applications:

- Language Modeling
- Speech Recognition

Advantages:

- Enhanced Context Understanding
- Improved Performance on Sequential Tasks

Uses of Transformer Models

Transformers are commonly used in NLP tasks, such as:

Language translation

Transformers efficiently translate text from one language to another while preserving contextual meaning.

Text classification

They can categorize or classify text into predefined groups based on content.

Uses of Transformer Models

Transformers are commonly used in NLP tasks, such as:

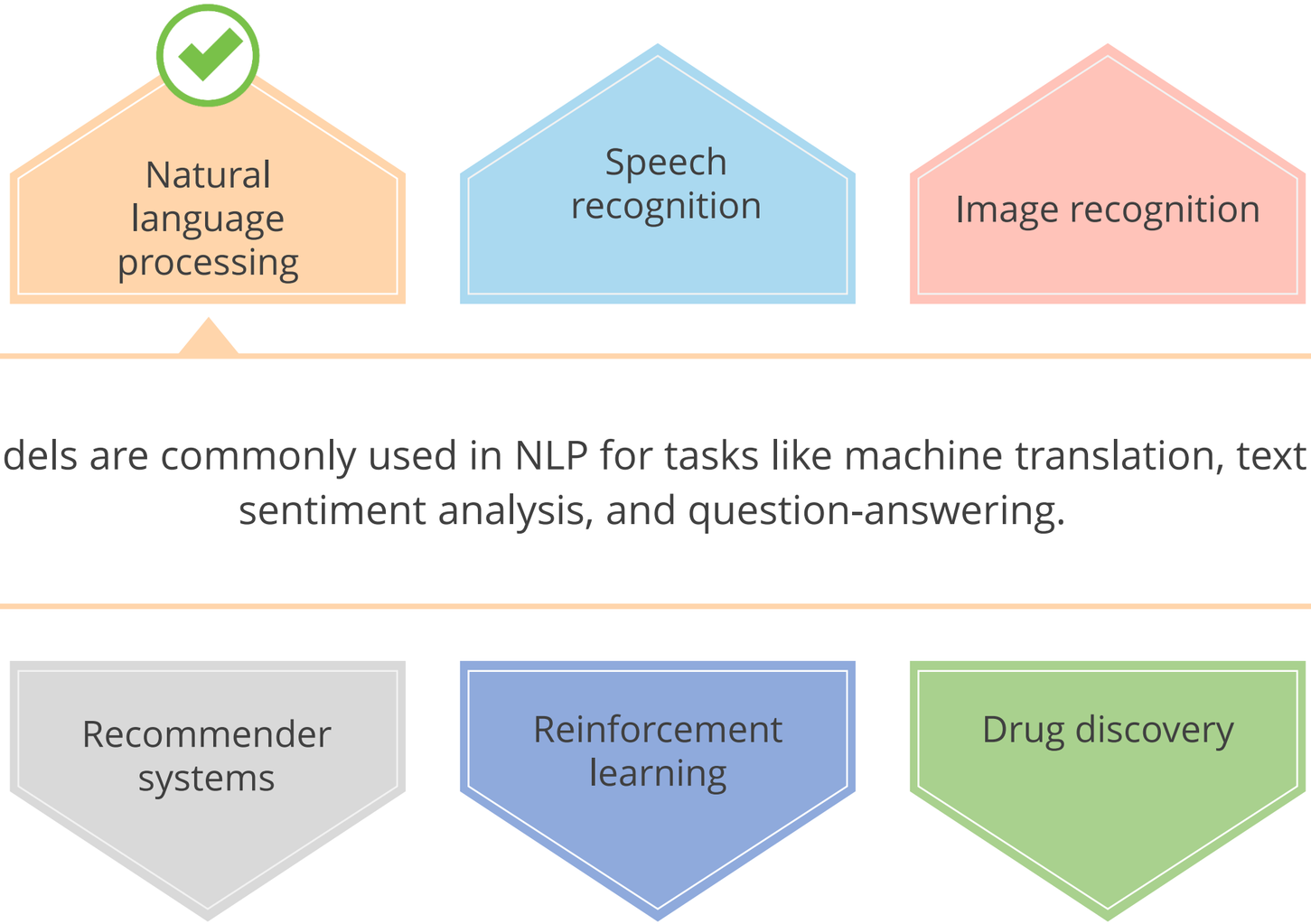
Language generation

They can generate human-like, contextually relevant text.

Question answering

Transformers can parse context to provide accurate answers to specific questions in natural language processing.

Applications of Transformer Models



Applications of Transformer Models

Natural language processing

Speech recognition

Image recognition

The transformer model has been employed in automatic speech recognition systems to convert spoken language into written text.

Recommender systems

Reinforcement learning

Drug discovery

Applications of Transformer Models

Natural language processing

Speech recognition

Image recognition

The transformer model has been adapted for computer vision tasks, such as image classification, object detection, and image captioning.

Recommender systems

Reinforcement learning

Drug discovery

Applications of Transformer Models

Natural language processing

Speech recognition

Image recognition

The transformer model has been applied in recommendation systems to provide personalized recommendations based on user preferences and behavior.

Recommender systems

Reinforcement learning

Drug discovery

Applications of Transformer Models

Natural language processing

Speech recognition

Image recognition

Transformer models are used in reinforcement learning to enhance decision-making in sequential tasks like game-playing and robot control.

Recommender systems

Reinforcement learning

Drug discovery

Applications of Transformer Models

Natural language processing

Speech recognition

Image recognition

Transformer models are utilized in drug discovery to predict molecular properties, design new molecules, and speed up the drug development process.

Recommender systems

Reinforcement learning

Drug discovery

Discussion: Transformer Models

Duration: 10 minutes

- What are transformer models?

Answer: Transformer models are advanced neural network architectures widely used in natural language processing tasks, offering superior performance in areas such as machine translation, text generation, and sentiment analysis.

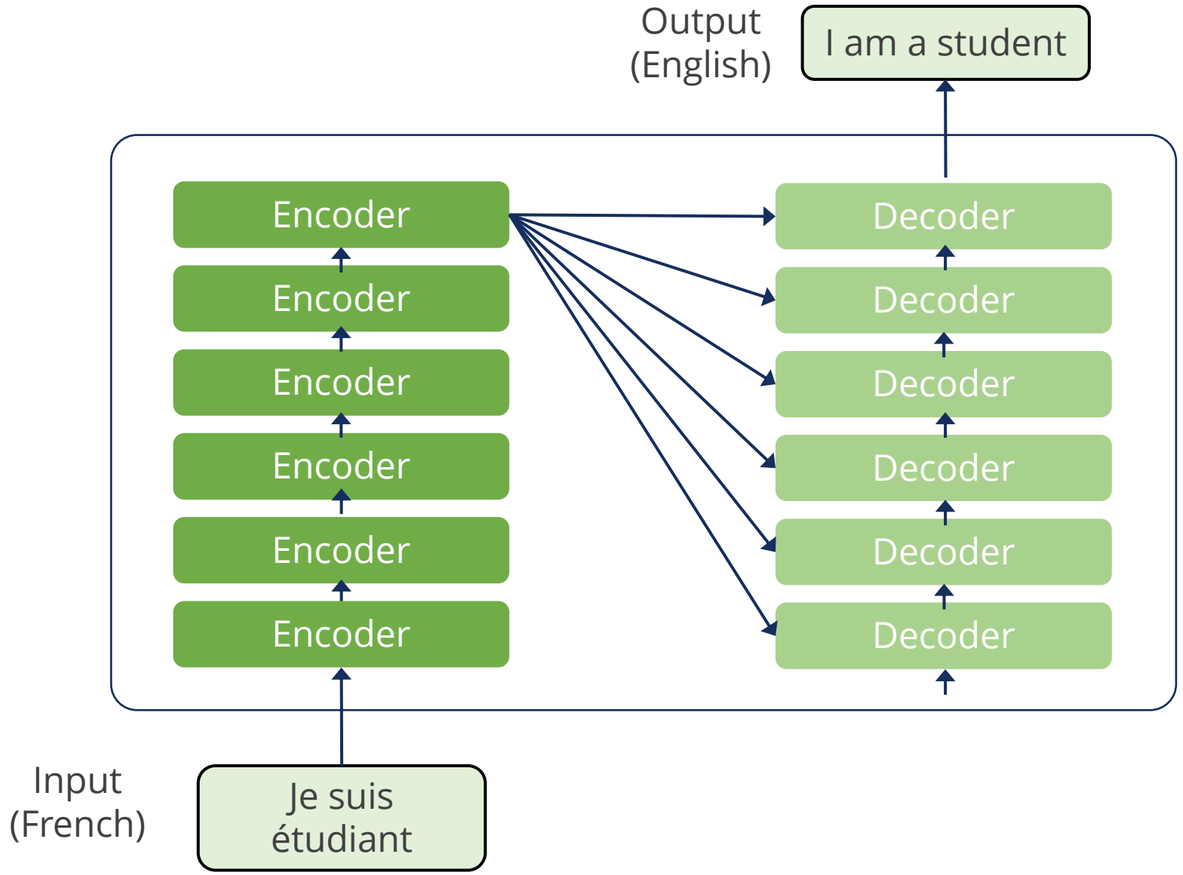
- What are some applications of transformer learning?

Answer: Transformer learning is used in machine translation, natural language processing, speech recognition, image captioning, recommendation systems, document summarization, and generative modeling.

Architecture of the Transformer Model

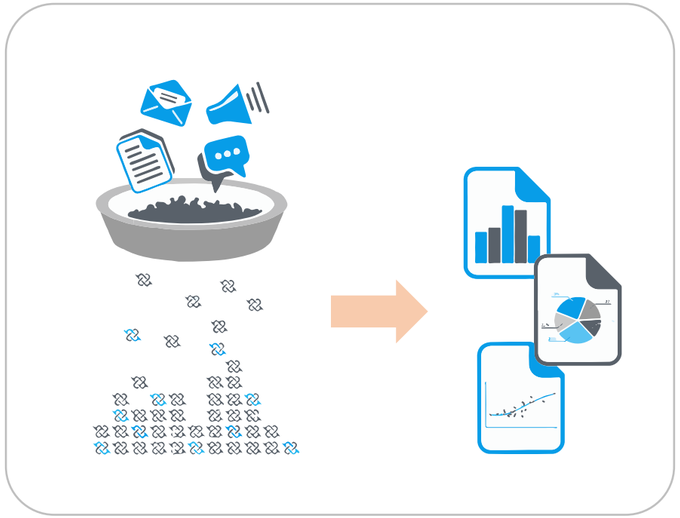
Transformer Model Architecture

At a high level, a transformer consists of an encoding and a decoding component.



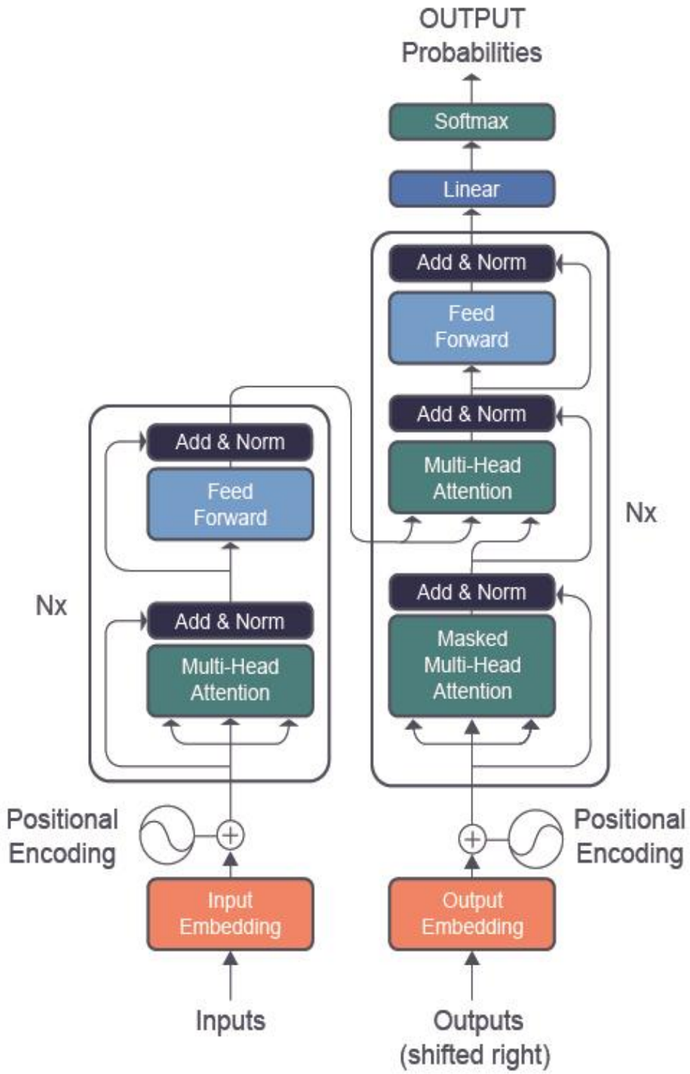
Transformer Model Architecture

- Encoding processes input sequences and captures contextual information.
- Decoding generates an output sequence and predicts the next word based on context.
- Architecture ensures accurate and coherent sequence processing and generation.



Transformer Model Architecture

It involves stacking multiple identical encoders and decoder layers for sequence processing.



- In a real transformer model, it is common to have six or more of these identical layers.
- The stacking of these layers enables effective information capture and generation in transformer models.

Transformer Model Architecture

Positional encoding and stacked encoders or decoders propel the input toward intelligent decision-making.

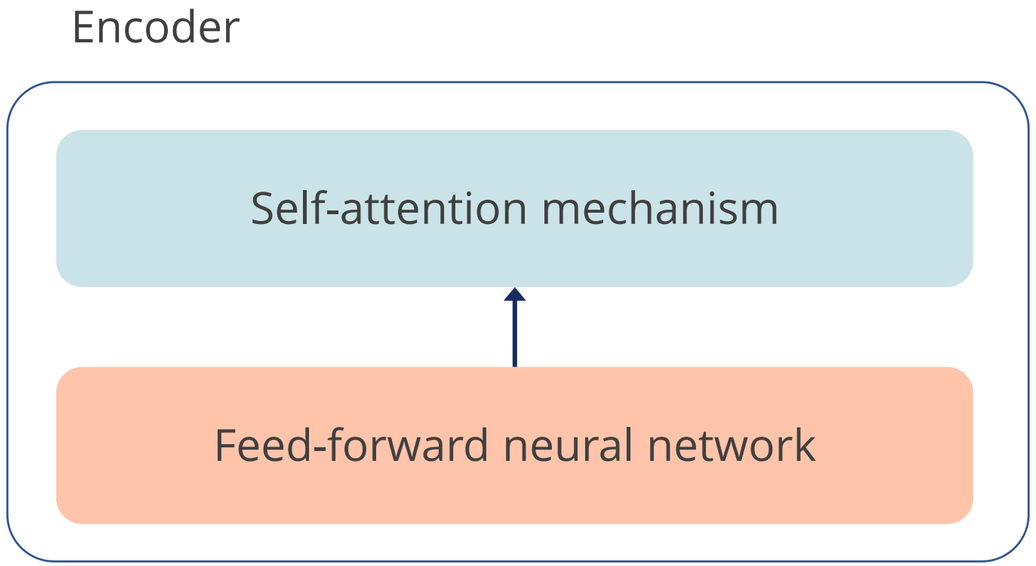
Input: Je suis étudiant

After embedding the input, it undergoes positional encoding, which involves adding a vector to it.

Next, it proceeds through a series of stacked encoders and decoders, each of which is identical in number.

Transformer Model Architecture

Each encoder has two layers:



Transformer Model Architecture

The general flow and components are:

Embedded vector: Initial input representation through an embedding layer.

Encoders: Multiple layers processing the embedded input.

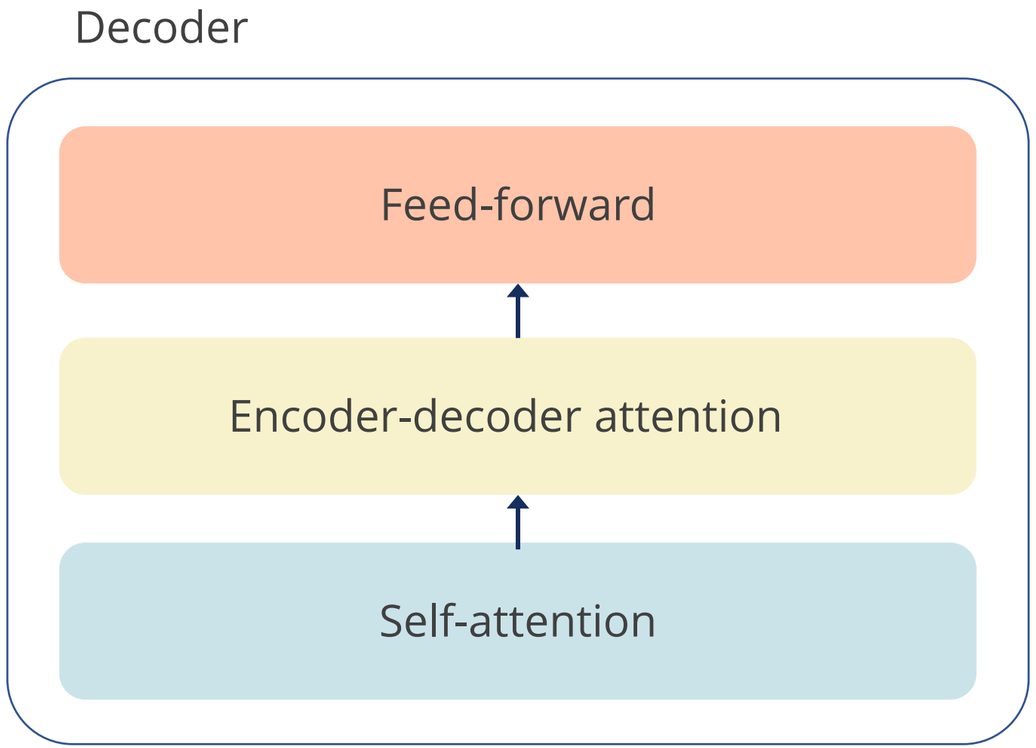
Self-attention layer: Captures dependencies and context in each encoder.

Feedforward neural network: Enhances pattern capture in encoders.

Final encoder output: Representation used for translation or generation tasks.

Transformer Model Architecture

A decoder has the following layers:



Transformer Model Architecture

The encoder-decoder attention layer helps the decoder focus attention on specific words.

The self-attention layer helps the model examine other words within the input to comprehensively understand and encode each individual word.

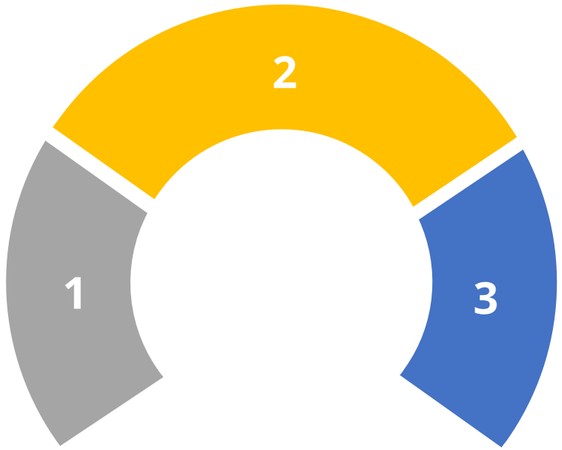
It is an attention mechanism that relates different positions of a single sequence to compute its proper representation.

Transformer Model Architecture

The self-attention layer calculates three vectors from each encoder’s input vector, such as:

Key Vector (K): Scores words to be attentive to.

Query Vector (Q): Scores words based on attention needs.



Value Vector (V): Represents actual word content for output.

Transformer Model Architecture

During the training phase, vectors are trained and updated iteratively.

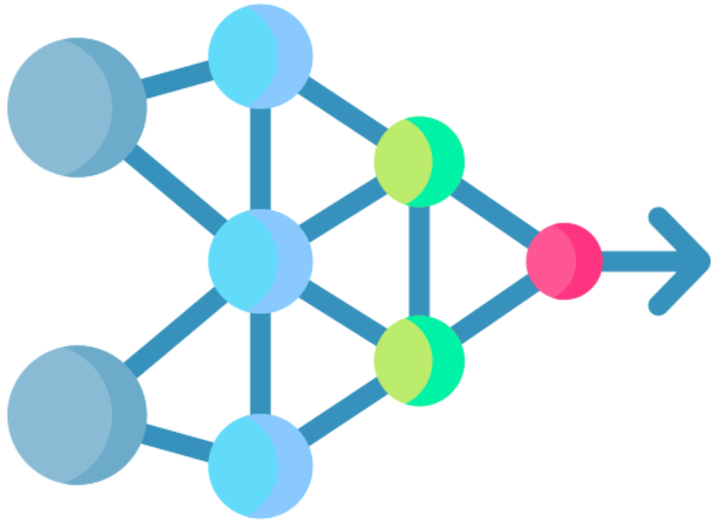
The self-attention score for each input word is defined by the following equation:

$$\text{Attention (Q,K,V)} = \text{Softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)\text{V}$$

The process is repeated eight times independently in parallel with different weights and then aggregated, which is called multi-head attention.

Transformer Model Architecture

The final output of the last decoder is passed through a linear layer.



This layer is a fully connected neural network that transforms the decoder output vector into a significantly larger vector known as a logit vector.

Transformer Model Architecture

The SoftMax layer converts the logits into a probability distribution, crucial for determining the most likely output.

OUTPUT: I am a student

Introduction to BERT Model

Discussion

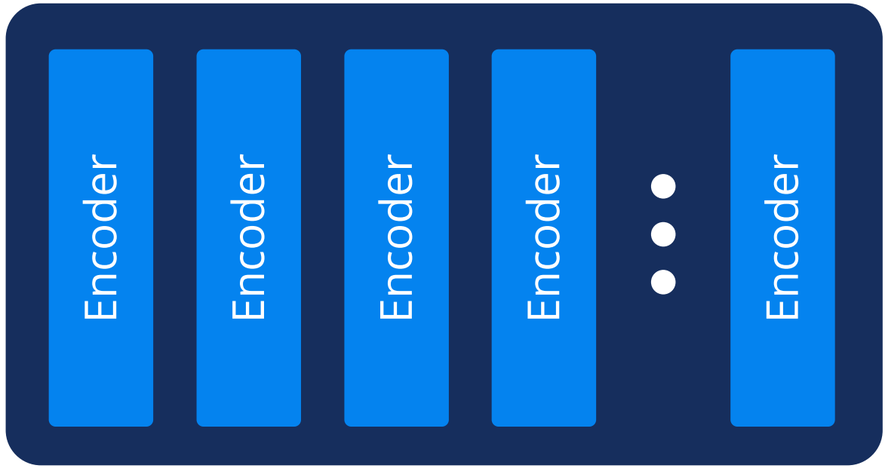
Discussion: BERT Model

Duration: 10 minutes

- What is the BERT model?
- What are some applications of the BERT model?

BERT Model

BERT stands for Bidirectional Encoder Representations Transformer.



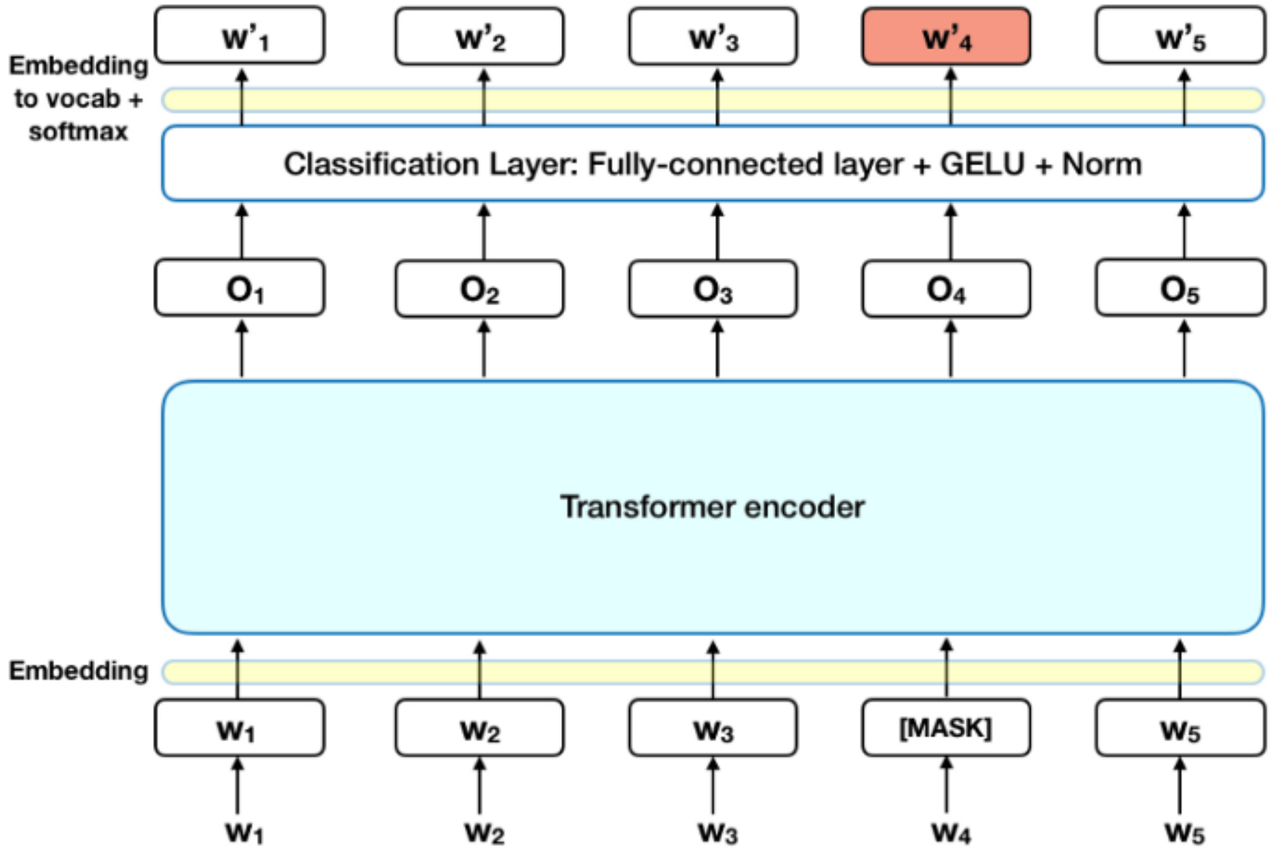
It is a transformer model without any decoder modules and only has a trained encoder stack.

The transformer encoder tends to read the entire sequence of words at once.

It learns the context of the given input rather than learning it in sequence. It is called contextual learning.

BERT and Masked Language Modeling

A Masked Language Model (MLM) is a type of language model used in the pre-training of models like BERT (Bidirectional Encoder Representations from Transformers). The working of an MLM is as shown below:



BERT and Masked Language Modeling

In Masked Language Modeling (MLM), 15 percentage of the words are replaced with a MASK token before feeding the sequence of words into BERT.

The model attempts to predict the original values of the masked words by leveraging the contextual information available.

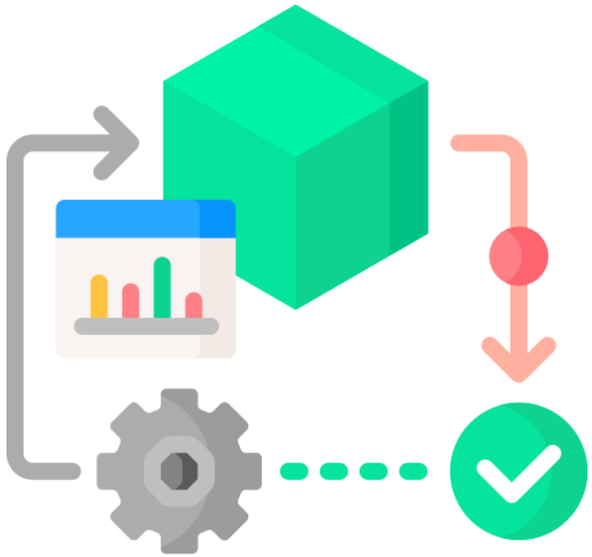
BERT and Masked Language Modeling

MLM performs the following:

- A percentage of the input tokens are randomly masked out in the input sequence.
- BERT encoder's masked output token is transformed to predict the probability of each word in the vocabulary.
- SoftMax is used to convert these values into probabilities, indicating the model's confidence in its word predictions.

BERT and Masked Language Modeling

The BERT loss function considers only predictions related to the masked values, disregarding other outputs.



The model converges slower than directional models.

Use Cases for BERT

BERT is widely used for the following tasks:

Text classification: It identifies text characteristics, like fraud detection.

Text generation: It generates text, specifically chatbot responses.

Use Cases for BERT

BERT is widely used for the following tasks:

Search engine optimizations: It improves search relevance for user queries.

Question-answering (Q&A) systems: It helps in accurate Q&A responses.

Discussion: BERT Model

Duration: 10 minutes

- What is the BERT model?

Answer: BERT is a cutting-edge deep learning model for natural language processing tasks, based on the Transformer architecture. It considers the entire context of a word in a sentence and has achieved remarkable success in various NLP tasks.

- What are some applications of the BERT model?

Answer: BERT has been applied in sentiment analysis, named entity recognition, question answering, text classification, natural language understanding, machine translation, and document classification.

Assisted Practices

Let’s understand the concept of BERT and its text classification using Jupyter Notebooks.

- 11.05_Introduction_to_BERT__V3
- 11.06_Text_Classification_using_BERT

Note: Please refer to the Reference Material section to download the notebook files corresponding to each mentioned topic

ASSISTED PRACTICE

Key Takeaways

- Transformers are known for their long-range memory dependencies and acquire this functionality through the concept of self-attention.
- A transformer consists of an encoding component and a decoding component.
- BERT is a transformer model but without any decoder module and only has a trained encoder stack.
- In an MLM, 15% of the words are replaced with a MASK token before feeding the sequence of words into BERT.

Knowledge Check

Knowledge Check

1

What are the components of a transformer model?

A. Encoding and decoding

B. Classification and regression

C. Activation and dropout

D. Convolution and pooling

Knowledge Check 1

What are the components of a transformer model?

A. Encoding and decoding

B. Classification and regression

C. Activation and dropout

D. Convolution and pooling

The correct answer is **A**

The transformer model consists of an encoding component consisting of stacks of encoders and a decoding component consisting of stacks of decoders.

Knowledge Check 2

What is BERT?

A. A model that solves the problem of long-range dependencies in sentences

B. A language model based on the transformer architecture

C. A feedforward neural network

D. None of the above

Knowledge Check 2

What is BERT?

A. A model that solves the problem of long-range dependencies in sentences

B. A language model based on the transformer architecture

C. A feedforward neural network

D. None of the above

The correct answer is **B**

BERT is a language model based on transformer architecture.

Knowledge Check

3

What is the output of the BERT classifier?

A. A probability distribution over the pre-defined categories

B. A transformed input vector

C. A masked language model

D. None of the above

Knowledge Check

3

What is the output of the BERT classifier?

A. A probability distribution over the pre-defined categories

B. A transformed input vector

C. A masked language model

D. None of the above

The correct answer is **A**

The output of the BERT classifier is a probability distribution over the pre-defined categories.

Thank You!