Bert and GPT4 Architectures

BERT Architectures (Bidirectional Encoder Representations from Transformers):

transformer-based model developed by Google that has significantly advanced natural language processing (NLP). Its architecture is designed to understand the context of words in a sentence more effectively than previous models.

BERT is built entirely on the **Transformer encoder** architecture, which consists of multiple layers of attention and feed-forward neural networks. It does not use the Transformer decoder part since BERT focuses solely on understanding (not generating) text.

Key elements of BERT's architecture:

1. **Transformer Encoder Layers**: Typically, 12 layers for BERT-Base and 24 for BERT-Large.
2. **Attention Heads**: 12 for BERT-Base and 16 for BERT-Large.
3. **Hidden Units**: 768 for BERT-Base and 1024 for BERT-Large.
4. **Parameters**: About 110 million for BERT-Base and 340 million for BERT-Large

Components of BERT Architecture:

1. Input Representation

- BERT's input representation combines:
- Token Embeddings: Convert words into fixed-size vectors.
- Segment Embeddings: Indicate whether a token belongs to sentence A or B (used in tasks with sentence pairs).

Position Embeddings: Capture word order by adding a unique vector for each position.

Input Format:

- [CLS] Sentence A [SEP] Sentence B [SEP]
- [CLS]: Special token for classification tasks.
- [SEP]: Separates sentences or marks sentence end

2. Bidirectional Attention

- Unlike traditional models that process text sequentially (left-to-right or right-to-left), BERT uses bidirectional self-attention. This approach:
  - Allows each word to attend to every other word in the input, both forward and backward.
  - Captures richer context and relationships between words.

3. Self-Attention Mechanism

- Multi-Head Attention: Each layer has multiple self-attention heads to focus on different parts of a sentence simultaneously.
- Scaled Dot-Product Attention: Computes attention scores between words to determine their importance.

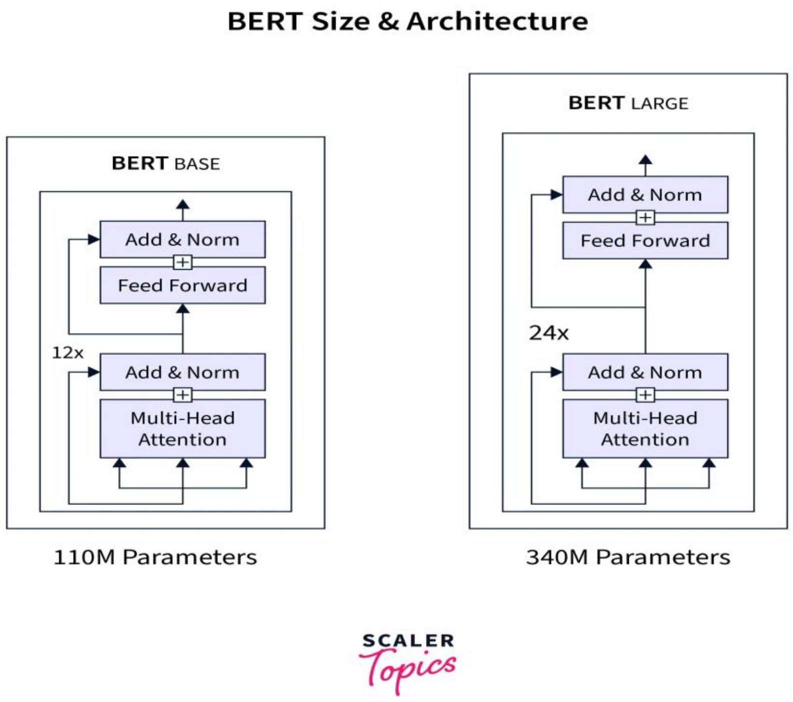
4. Feed-Forward Neural Networks

- Each encoder layer includes a fully connected feed-forward network, applied independently to each position.

5. Pretraining Tasks

- BERT’s power comes from its pretraining tasks:
  - Masked Language Modeling (MLM):
    - Randomly masks 15% of the input tokens.
    - The model predicts the masked tokens based on surrounding context.
- Next Sentence Prediction (NSP):
  - Trains the model to understand sentence relationships.
  - 50% of the time, the second sentence follows the first logically; 50% of the time, it is random.

BERT Architecture:



GPT-4 Architecture (Generative Pre-Trained Transformer 4):

GPT-4 is a transformer-based model developed by OpenAI that has pushed the boundaries of natural language processing (NLP). Its architecture is designed to generate coherent, context-aware text by predicting the next word in a sequence. Unlike BERT, which focuses on understanding text through bidirectional attention, GPT-4 is optimized for text generation using a unidirectional approach.

GPT-4 is built entirely on the **Transformer decoder architecture**, which consists of multiple layers of masked self-attention and feed-forward neural networks. It does not use the Transformer encoder part since GPT-4 is focused solely on generating (not understanding) text bidirectionally

Key Elements of GPT-4's Architecture:

1. **Transformer Decoder Layers**: Typically includes 48 layers for GPT-4 (compared to 12 for GPT-2 and 96 for some enhanced versions).
2. **Attention Heads**: Can range up to 128, allowing the model to focus on different parts of the input sequence.
3. **Hidden Units**: Varies but can include up to 12,288 dimensions for GPT-4's large-scale versions.
4. **Parameters**: Estimated to be in the range of hundreds of billions, significantly larger than GPT-3's 175 billion parameters.

Components of GPT-4 Architecture:

1. Input Representation

- Token Embeddings: Converts words or subwords into fixed-size vectors using a Byte-Pair Encoding (BPE) tokenizer.
- Position Embeddings: Adds unique vectors to tokens based on their position in the input to capture word order.
- Input Format:
    - A sequence of tokens with no special [CLS] or [SEP] tokens, as used in BERT.
    - Unidirectional context, where each token can only attend to previous tokens.

2. Unidirectional Attention

- Causal Masking: Ensures that tokens can only attend to previous tokens in the sequence, preventing future information leakage during generation.
- Context-Aware Generation: Builds understanding sequentially, generating one token at a time based on prior tokens.

3. Self-Attention Mechanism

- Multi-Head Attention: Each layer has multiple self-attention heads to capture different aspects of the input sequence simultaneously.
- Masked Self-Attention: Prevents a token from attending to future tokens by applying a causal mask during attention score computation.
- Scaled Dot-Product Attention: Computes attention scores to determine the importance of tokens relative to one another.

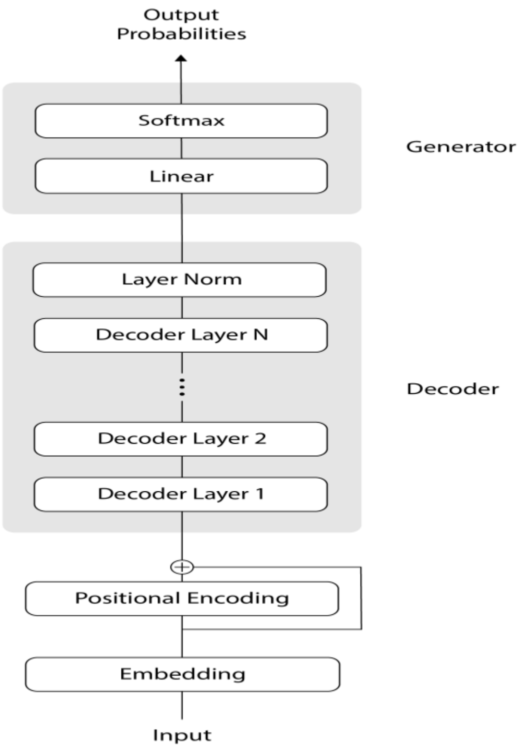
4. Feed-Forward Neural Networks

- Each decoder layer includes a fully connected feed-forward network applied independently to each position, enhancing the model’s ability to transform information between layers.

5. Pretraining Tasks

- GPT-4’s power comes from its pretraining objective:
- Next Token Prediction (Autoregressive Pretraining):
- Trains the model to predict the next token in a sequence based solely on previous tokens.
- Encourages the model to learn long-term dependencies and context in text.

GPT4 Architecture:



Compression between BERT and GPT-4:

	BERT	**GPT-4**
Model Type	Encoder only	Decoder Only
Directional	Bidirectional	Direction
Pre-training Objective	Masked language modeling (MLM)	Autoregressive (casual) language modeling
Fine Tuning	Task-specific layer added on top of the pre-trained BERT model	Providing task-specific prompts using few-shot or one-shot adaptation and adapting the model's parameters
Use Case	Sentiment Analysis Named entity Recognition Word Classification	Text generation Text completion Creative writing
Original Organizations	Google AI	OpenAI