📌 Attention Mechanism

1️⃣ Why is Attention Needed in Sequence Models?

✅ Problem with Traditional Sequence Models (LSTMs, Bi-LSTMs)

- LSTMs process **sequential data** step by step.
- For long sequences, **early information fades**, making it hard to capture **long-range dependencies**.
- Example: In **Machine Translation**, the model needs to relate words **at the start of a sentence** with **words at the end**.

✅ How Attention Solves This?

- Instead of relying only on the **final hidden state**, **Attention dynamically decides** which past words are important at each step.
- It assigns **higher importance (weights)** to relevant words, improving **context understanding**.

2️⃣ Types of Attention: Soft vs. Hard Attention

Feature	Soft Attention	Hard Attention
Weight Distribution	Assigns different weights to all words	Selects only **one** most relevant word
Differentiability	Fully differentiable (trainable with gradient descent)	Non-differentiable (uses reinforcement learning)
Computation	Efficient	Expensive
Use Case	Most NLP models (Translation, Summarization, NER)	Vision-based models (Image Captioning)

✅ **Most NLP models use Soft Attention** since it allows smooth learning with **gradient updates**.

3️⃣ Scaled Dot-Product Attention (Core of Transformers)

✅ Key Idea:

Instead of treating **all words equally**, compute the **similarity** between the current word and all past words.

✅ Formula for Attention Score:

Attention(Q,K,V)=softmax(QK^T / dk^1/2)V

Where:

- **Q (Query):** Word we are focusing on.
- **K (Key):** All words in the sequence.
- **V (Value):** Word representations.
- **dk (Scaling factor):** Prevents extremely large values, stabilizing gradients.

✅ **Key Benefit:** Instead of treating **all past words equally**, it **learns** what to focus on dynamically.

4️⃣ Self-Attention Mechanism (Key, Query, Value Concept)

✅ How Does Self-Attention Work?

Each word **interacts** with all other words and decides **how important they are**.

Step-by-Step Process:

1. Convert input words into **vectors**.
2. For each word, compute:
   - **Query (Q)**: What am I looking for?
   - **Key (K)**: What do I contain?
   - **Value (V)**: What information should I pass?
3. Compute **attention scores** using dot-product similarity.
4. Multiply scores with **V** (Values) to get weighted output.
5. Pass the result to the next layer.

✅ Example (Sentence: "The cat sat on the mat.")

- If the current focus is **"sat"**, attention may assign:
  - **High weight** to "cat" (subject of the sentence).
  - **Lower weight** to "mat" (less relevant).

🔥 Summary

- **Attention Mechanism** helps Bi-LSTMs focus on important words in long sequences.
- **Types of Attention:** Soft (most NLP tasks) vs. Hard (image-based tasks).
- **Scaled Dot-Product Attention** improves efficiency using **Q, K, V matrices**.
- **Self-Attention** assigns dynamic weights to past words **based on relevance**.
- **Implemented in Bi-LSTMs** to improve performance in NLP tasks like **NER, Sentiment Analysis, Translation, and Summarization**.