**Matrix in Model Output**

**🔎 Step 1: Input to Model**

Suppose you feed this text:

This is a sample sentence for tokenization and encoding

Tokenized (BERT-base-uncased):

[CLS] this is a sample sentence for token ##ization and encoding [SEP]

That’s **12 tokens total**.

**🔎 Step 2: Model Processing**

* Each token ID is mapped to a **vector of size 768** (for bert-base-uncased).
* Then the Transformer layers (attention + feed-forward) refine these vectors in context.

**🔎 Step 3: Model Output = a Matrix**

* Shape = **(sequence\_length, hidden\_size)**.
* For your example:

12×768

* Each row corresponds to **one token** (including [CLS] and [SEP]).
* Each column corresponds to one dimension of that token’s **final hidden representation**.

**📊 Example Matrix Layout**

[

[ 0.12, -0.03, ..., 0.44 ], ← [CLS]

[ 0.21, 0.77, ..., -0.11 ], ← "this"

[ 0.09, 0.65, ..., 0.33 ], ← "is"

...

[ 0.14, -0.22, ..., 0.56 ] ← [SEP]

]

**🔎 Step 4: How this Matrix is Used**

* **[CLS] row** → used for classification tasks (e.g., sentiment, intent).
* **All token rows** → used for token-level tasks (e.g., NER, QA, translation).
* You can also pool or average across tokens to get a single sentence embedding.

✅ So:  
The **matrix in model output** is simply:  
**one vector per token (rows), each of dimension = hidden size (columns).**