Pre-Processing Step with Python: Explanation

October 4, 2024

Overview

The pre-processing step in the Python script `run_inference.py` is used to generate float values that are compatible with the STM32 model. This step is necessary because the TensorFlow `Tokenizer` and `Embedding` layers cannot be used directly in the STM32 X-CUBE-AI framework due to resource constraints. Instead, we use a simplified manual mapping in Python, which converts text inputs to numerical values. Below, we outline how this process works and why it's required for our STM32 deployment.

1. Running the Python Script

The Python script processes each text input (e.g., "The movie was great!") and maps it to a corresponding float value:

- The script utilizes a basic vocabulary list, where each word is assigned a numerical value.
- Instead of generating token IDs like the TensorFlow tokenizer, it directly outputs pre-determined float values (e.g., `0.8`, `0.5`, `0.2`).
- This is done to simplify the input format, making it easy to feed into the STM32 model.

2. Simplified Vocabulary

The vocabulary used in the Python script is a reduced set of words, each mapped to a specific sentiment value:

- For example:
  - `"The movie was great!"` → `0.8`
  - `"The movie was okay."` → `0.5`
  - `"The movie was terrible..."` → `0.2`
- These float values are chosen to correspond to the sentiment scores directly, avoiding the need for complex embeddings.

3. Why the TensorFlow Tokenizer and Embeddings Are Not Used

In the original TensorFlow tutorial, a `Tokenizer` is used to convert words into a sequence of integers, and an `Embedding` layer transforms these integers into dense vectors:

- This process is computationally expensive and memory-intensive, making it unsuitable for microcontrollers like the STM32.
- The X-CUBE-AI framework does not support embeddings natively, which means these layers cannot be used directly in the STM32 implementation.

Instead, the Python script handles this complexity by reducing the text to float values, allowing us to bypass the need for embeddings and keep memory usage low.

4. Using Preprocessed Floats in STM32

The preprocessed float values generated by the Python script are used as direct inputs to the STM32 model:

- These values are fed into the first Dense layer of the model without any additional tokenization or embedding.
- This approach reduces complexity and ensures that the input format is compatible with the dense-only architecture of the deployed model.
- The outputs of the STM32 model can then be compared against the outputs of the Python model to verify parity and ensure that the simplified input format produces consistent results.

Conclusion

The pre-processing step using Python is an essential part of the workflow for deploying the text classification model on STM32. It bridges the gap between the original TensorFlow model and the simplified STM32 implementation, ensuring that inputs are correctly formatted and compatible with the hardware constraints of the microcontroller.