# Using bi-encoder in multi class classification

Compiled with Google Gemini, 5/12/2025

To use a bi-encoder in multi-class classification, you can employ a strategy that leverages the bi-encoder's efficiency in retrieval and then use a cross-encoder (or another classifier) for more accurate ranking or prediction. This involves encoding each class label separately and then using the bi-encoder to retrieve relevant examples for each class, followed by a cross-encoder or a final classifier to determine the most accurate classification.

Here's a more detailed breakdown:

1. **Encoding Class Labels:** For each of the N classes, use the bi-encoder to generate a separate vector representation for that class label.
2. **Encoding Input Data:** Encode your input data (e.g., text, images) using the same bi-encoder.
3. **Retrieval for Each Class:** For each class, use the bi-encoder to retrieve the most similar examples (from your training or a larger dataset) based on the class representation and the input data representation.
4. **Cross-Encoder or Final Classifier:** Use a cross-encoder (or another classification model) to analyze the retrieved examples and the input data, and predict the most likely class.
5. **Output:** The final classifier outputs the predicted class for the input data.

Why this approach is useful:

* **Efficiency:**

Bi-encoders are efficient at encoding and retrieving similar items, making them suitable for handling large datasets.

* **Accuracy:**

Cross-encoders or other classifiers can then refine the results, potentially achieving higher accuracy than relying solely on a bi-encoder.

* **Scalability:**

This approach can be scaled to handle a large number of classes by leveraging the bi-encoder for efficient retrieval and the cross-encoder for more accurate prediction.

Example (Information Retrieval Scenario):

Imagine you have a set of documents (your input data) and you want to classify them into different topics (your classes). You could use a bi-encoder to:

1. Encode each topic as a vector.
2. Encode each document as a vector.
3. Retrieve the most relevant documents for each topic.
4. Use a cross-encoder or a classifier to determine the final classification of each document.

References

[1] [Bi-Encoders and Cross-Encoders: Two Sides of the Retrieval Coin, Marc Puig, 2024](https://medium.com/@mpuig/bi-encoders-and-cross-encoders-two-sides-of-the-retrieval-coin-06a95fe18619)

[2] [Cross-Encoders vs Bi-Encoders: A Deep Dive into text encoding methods, R. Bhatia, 2024](https://medium.com/@rbhatia46/cross-encoders-vs-bi-encoders-a-deep-dive-into-text-encoding-methods-d9aa890d6ca4)

[3] Cross-Encoders on SBert.net: <https://www.sbert.net/examples/cross_encoder/applications/README.html>

[4] Sentence Transformers on SBert.net: <https://sbert.net/docs/sentence_transformer/usage/usage.html>