**FAISS Indexing Concepts and Index Types**

**What is Indexing in FAISS?**

In similarity search tasks, **indexing** refers to creating a data structure that enables fast retrieval of vectors similar to a given query vector, based on a distance metric (e.g., Euclidean distance or cosine similarity).

**FAISS** (Facebook AI Similarity Search) is a library designed to perform efficient similarity search and clustering on dense vectors, especially at large scale.

Instead of brute-force comparing every vector, FAISS builds indexes to achieve:

* Fast nearest neighbor search
* Balance between speed, memory, and accuracy

**Main FAISS Index Types**

**1. Flat (Brute-force) Index**

* **Description:** Stores all vectors exactly without compression or approximation.
* **Examples:**
  + faiss.IndexFlatL2(d) — L2 (Euclidean) distance
  + faiss.IndexFlatIP(d) — Inner product (for cosine similarity if vectors are normalized)
* **Pros:** Exact results
* **Cons:** Slow for large datasets

**2. IVF (Inverted File Index)**

* **Description:** Partitions the dataset into clusters using k-means (number of clusters = nlist). Searches only a subset of clusters (nprobe clusters).
* **Example:**
  + faiss.IndexIVFFlat(quantizer, d, nlist)
* **Notes:** Requires training (index.train(data)) before adding vectors.
* **Pros:** Faster than flat for large datasets
* **Cons:** Approximate results; quality depends on cluster parameters

**3. PQ (Product Quantization)**

* **Description:** Compresses vectors into compact codes for memory-efficient storage.
* **Example:**
  + faiss.IndexPQ(d, m, nbits)
* **Use Case:** Large-scale datasets (millions to billions)
* **Pros:** Very low memory usage
* **Cons:** Slower indexing/search than IVF; approximate

**4. IVF + PQ (Inverted Index with Quantization)**

* **Description:** Combines IVF clustering and PQ compression.
* **Example:**
  + faiss.IndexIVFPQ(quantizer, d, nlist, m, nbits)
* **Pros:** Balanced speed, accuracy, and memory use
* **Cons:** Requires training; approximate

**5. HNSW (Hierarchical Navigable Small World)**

* **Description:** Graph-based approximate nearest neighbor method.
* **Example:**
  + faiss.IndexHNSWFlat(d, M)
* **Pros:** Very fast search, no training needed
* **Cons:** Sensitive to parameter M controlling graph connectivity

**6. Binary Indexes**

* **Description:** Specialized for binary vectors (e.g., ORB, BRIEF descriptors).
* **Example:**
  + faiss.IndexBinaryFlat(d)
* **Use Case:** Binary feature similarity search with Hamming distance

**Additional Tools**

* **IndexPreTransform:** Apply transformations (PCA, normalization) before indexing.
* **IndexIDMap:** Maps FAISS internal IDs to custom user-defined IDs.

**Choosing the Right Index**

| **Dataset Size** | **Accuracy Needed** | **Recommended Index** |
| --- | --- | --- |
| Small (<100k) | High | IndexFlatL2 |
| Medium (100k-1M) | Medium | IndexIVFFlat |
| Large (>1M) | Medium-Low | IndexIVFPQ or IndexPQ |
| Very Large (10M+) | Medium-Fast | IndexHNSWFlat |
| Binary Data | Varies | IndexBinaryFlat |