POLITECNICO DI TORINO
Telecommunications Engineering
Information Theory and Signal Processing Applications

Master Degree Dissertation

Nearest Neighbour Search using binary clustered Neural Networks

Applied to object retrieval and classification

Candidate
Demetrio Ferro

Advisor
Guido Montorsi

Internship Student at
Télécom Bretagne

TELECOM Bretagne

TÉLÉCOM BRETAGNE

Co-Advisor:

Claude Berrou

Internship Tutor:

Vincent Gripon

Télécom Bretagne

Brest, FRANCE

This work has been funded in part by the ERC under the EU Seventh Framework Programme (FP7 / 2007 - 2013) / Grant agreement n° 290901.

PRESENTATION OVERVIEW

PROBLEM STATEMENT

NEAREST NEIGHBOUR SEARCH

The problem of searching for the Nearest Neighbour is formulated as:

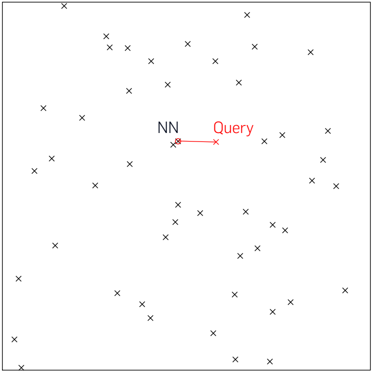
> "**Given a collection of data points and a query point in a hyper-dimensional metric space, find the data point that is closest to the query one.**" [Kevin Beyer, 1998]

The closeness metric is generally considered as **Euclidean Distance**.

Motivations can be found in its wide set of applications, mainly related to the domains of Data Processing and Machine Learning:

- → **Object Retrieval**
- → **Classification**
- Computational Statistics
- Pattern Recognition
- Computer Vision
- Data Mining.

PROBLEM STATEMENT



Example of Nearest Neighbour search query over a two-dimensional set of uniform distributed points.

Running an **Exhaustive Search**:

→ Computing Euclidean Distances,

→ Retrieving the minimum one.

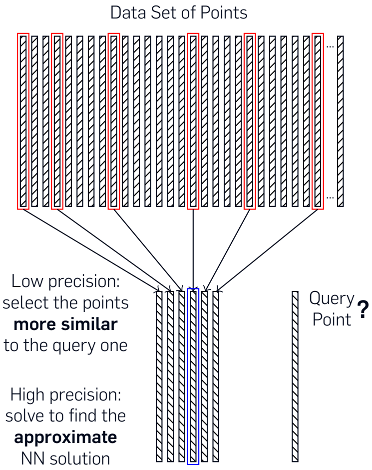
The **Computational Complexity** is:

→ Linear with the number of data points, considered to be N,

→ Linear with the dimensionality of the search space d.

It may hence be denoted as $\mathcal{O}(N \cdot d)$.

PROPOSED SOLUTION

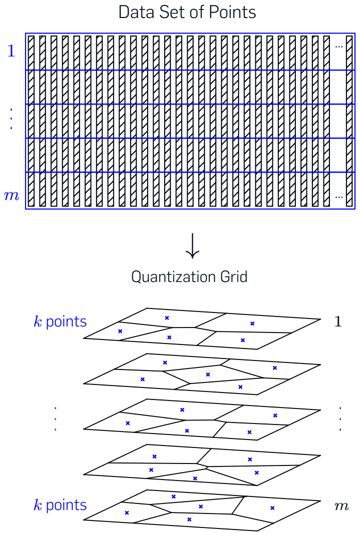


The proposed solution is structured as:

- **Training Stage**
  - → Coarse Vector Quantization acts on the data dimensionality,
  - → Neural Networks Learning allows a quick data access.
- **Query Stage**
  - → Neural Networks Polling acts on the data cardinality.
  - → Fine Vector Quantization Despite a higher compression rate, it works at low dispersion.

VECTOR QUANTIZATION

VECTOR QUANTIZATION

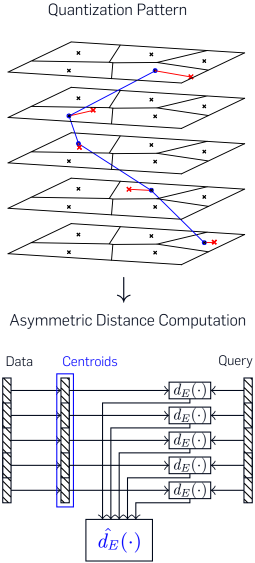


The quantization grid is computed empirically over the data set of points.

- **Splitting over m layers**
  - → The search space is split in contiguous orthogonal sections.
- **Clustering the layers to k points**
  - → Each data points section is processed to determine the best set of **cluster centroids**.

The optimality criteria considered is the minimum average dispersion one,
Its achievement is sought in the use of a specific procedure known as **k-means**.

PRODUCT QUANTIZATION



Product Quantization (PQ) is a state-of-the-art technique to solve Approximate NN (ANN) search:

- **Data Points Approximation**
  → Associate them centroids sets called **Quantization Patterns**.
- **Distance Metric Approximation**
  → The Euclidean Distance is computed as **Sum of Distances** over lower-dimensional layers.

The scan search can be accelerated by using a Look Up Table approach.

NEURAL NETWORKS

ARTIFICIAL NEURAL NETWORKS

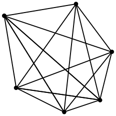
Artificial Neural Networks are a family of computational models able to accelerate the **data access** with a neuro-inspired approach.

Based on the biological model of **synaptic interaction**, it concerns a system of interconnected neural cells exchanging information messages.

The connections have numeric weights that can be tuned based on experience, making them capable of **learning** and **retrieving** messages.

They may be represented in two equivalent ways:

Weighted, Undirected Graphs



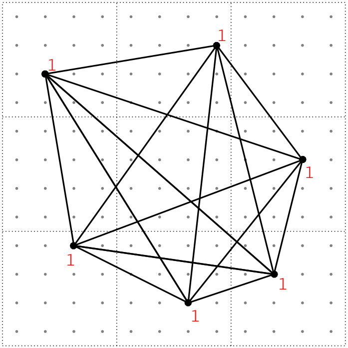
Nodes represent neurons, whereas edges are their interconnections.
They always generate fully connected polyhedra.

Connections Matrices

$$\begin{matrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \end{matrix}$$

Each of the elements expresses the activation of a cell, hence the weight of the connections between couple of nodes.

ADOPTED MODEL



Peculiar properties of the adopted model, can be expressed by:

→ **Binary Connections Weights**
Ones in the matrix establish active nodes, i.e. they identify only the existence of edges among nodes.

→ **Clustered neural activations**
The connections matrix is partitioned in such a way that only unique activations occur within each cluster.

The activation co-occurrences within clusters identify a fully connected sub-graph, or **Connections Pattern**.

FUNCTIONALITIES

Learning Rule

Considering $\mathbf{z}$ to be one of the **binary messages** in the **learning set** $\mathcal{Z}$ with a clustered structure, the **adjacency connections matrix** $W(\mathcal{Z})$ is generated as:

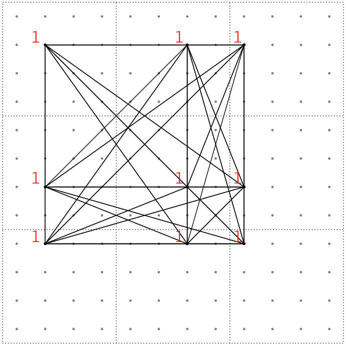
$$W(\mathcal{Z}) = g\left[\sum_{\mathbf{z}\in\mathcal{Z}} \left(\mathbf{z}\cdot\mathbf{z}^T\right)\right], \quad g(\xi) = \begin{cases} 1 & \text{if } \xi > 0 \\ 0 & \text{otherwise} \end{cases}, \; \xi \in \mathbb{Z}^*$$

Polling Rule

Being $\mathbf{z}_0$ a **binary query message**, still matching the clustering structure, the polling rule for the neural network trained by the set $\mathcal{Z}$ is expressed by the **score**:

$$s(\mathbf{z}_0, \mathcal{Z}) = \frac{\mathbf{z}_0^T \cdot W(\mathcal{Z}) \cdot \mathbf{z}_0}{(\mathbf{z}_0^T \cdot \mathbf{z}_0)^2}, \in [0,1] \subset \mathbb{R}$$

MOTIVATIONS OF USE



The choice of a binary, clustered model is motivated by many reasons:

→ Binary connection weights
Reduced **computational complexity** and preserve **memory occupancy**.

→ Clustered nodes structure
The learning process acts as **patterns overlapping**.
Clusters prevent from generating **false connections patterns**.

→ Adjacency connections matrix
For its **symmetry**, it carries a lot of **redundancy**, making it robust to **noise or memory faults**.

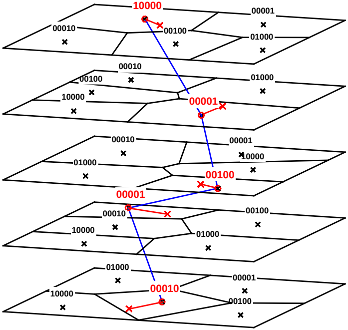
TRAINING STAGE

TRAINING STAGE OUTLINE

Set of operations executed offline, preparatory to the Query stage.

- Run a first **Coarse Vector Quantization**
  - → Build the Quantization Grids over m_c **layers**, each of them composed of k_c **centroids**, obtained by running k-means;
  - → Assign each data point to a **Quantization Pattern**.
- Go to the **Neural Networks Learning Stage**
  - → Convert the quantized data points to **binary messages**, so that they can be leant by binary neural networks.
  - → Identify L **learning sets** of binary messages to be learnt.
  - → applying the **learning rule** over all of the neural networks.

CLUSTERED BINARY LABELLING



$$\mathbf{z} = [\mathbf{10000\ 00001\ 00100\ 00001\ 00010}]$$

Quantization Patterns → Connections Patterns

Centroids Layers → Networks Clusters

It is performed a **Binary Labelling**:

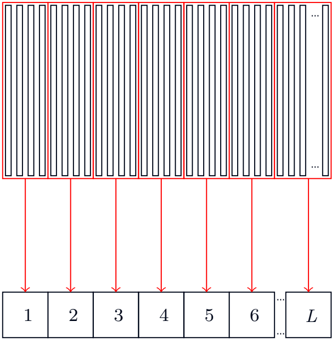
- → All of the centroids, over each layer are associated to **unitary Hamming weight**, from the canonical basis of order k.
- → For the network structure, only a **single activation** is allowed for each cluster.

Once all of the vectors are labelled, the whole search space is mapped to its binary version.

At this point, it is foreseen the use of a preprocessing step, aiming to partition the search space in a convenient way.

PREPROCESSING FOR ALLOCATION

Binary Labelled Data Set



Learning Sets

↓

Neural Networks: apply Learning Rule

$$W(\mathcal{Z}_l) = g\left[\sum_{\mathbf{z}\in\mathcal{Z}_l} (\mathbf{z}\cdot\mathbf{z}^T)\right], \quad \forall l \in [\![1, L]\!]$$

The binary labelled data set of points is partitioned in $\boldsymbol{L}$ **disjoint learning sets.**

It is used a greedy approach, aiming to:

→ **Portion of ones in the matrices**
Minimize the mutual Hamming distance within each set, hence the diversity of messages.

→ **Constant diversity**
The diversity of messages has to be uniformly distributed among the L learning sets $\mathcal{Z}_l, \forall l \in [\![1, L]\!]$.

→ **Constant cardinality**
The partition has to be fair, hence $|\mathcal{Z}_l| = n = N/L, \forall l \in [\![1, L]\!]$, in such a way that cardinality reductions do not depend on the specific query.

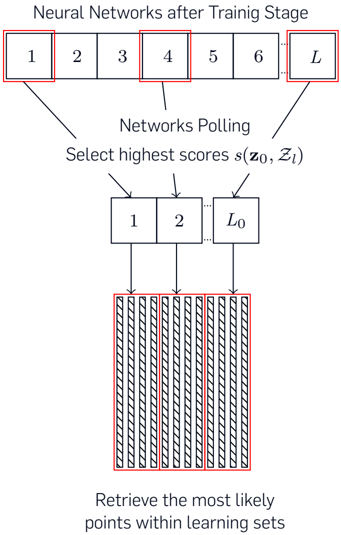
QUERY STAGE

QUERY STAGE OUTLINE

Set of operations executed on the run, when a query is performed.

- Resort to a **Neural Networks Polling**
  - → Compute the likelihood scores for all of the L networks;
  - → Select the **highest $\boldsymbol{L_0}$ scores** of likelihood;
  - → Retrieve the L_0 most likely **learning sets**.
- Apply a last **Fine Vector Quantization**
  - → Associated data points to a **fine** grid composed of $\boldsymbol{m_f}$ **layers**, with $\boldsymbol{k_f}$ **centroids** each;
  - → Approximate distance metrics with **reduced dispersion**.

NETWORKS POLLING



The selection of most likely sets is supposed to be performed as:

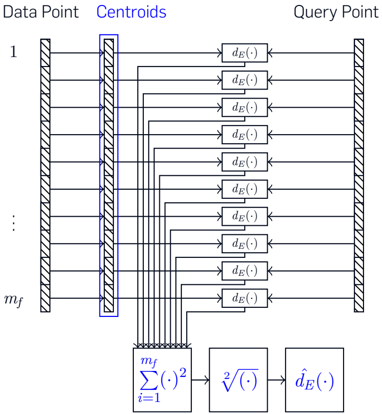
→ **Compute scores of likelihood**
For the query vector to be in the set $\mathcal{Z}_l$, learnt by the l^{th} neural network.

$$s(\mathbf{z}_0, \mathcal{Z}_l) = \frac{\mathbf{z}_0^T \cdot W(\mathcal{Z}_l) \cdot \mathbf{z}_0}{(\mathbf{z_0}^T \cdot \mathbf{z}_0)^2}, \ \forall l \in [\![1, L]\!]$$

→ **Select the highest L_0 scores**
That is, the vectors within the most likely learning sets.

Once the data points are retrieved, it is applied a better approximation to find the Nearest Neighbour with more precision.

FINE PRODUCT QUANTIZATION



Again, the quantization grid is computed empirically over the whole data set.

- → **Quantizing the data set**
  Each of the data points is associated to a fine quantization pattern.
- → **Distance Approximation**
  The distance among points is computed as the sum of distances over lower-dimensional sections.

To increase the reliability, it is foreseen the use of **multiple Recalls**.

It consists of iterative runs of the same technique, where solution obtained at each iteration is removed from the search space.

EMPIRICAL RESULTS

PERFORMANCES

→ **Metric of Evaluation**

The performances are measured in terms of success rate for **estimated solution to match the exhaustive one**, in a set of recalls of order r.

$$\eta_r = \begin{cases} 1 & \text{if } r\text{-th rank NN matches the exhaustive solution} \\ 0 & \text{otherwise} \end{cases}$$

→ **Computational Complexity**

Fine Product Quantization	Binary Clustered Networks
$\mathcal{O}(k_f d + \boxed{m_f N})$	$\mathcal{O}(k_f d + k_c d + \boxed{p_1 (m_c k_c)^2 L} + \boxed{m_f L_0 n})$
Distance Computation Fine Quantization	Coarse, Fine Quantization Scores Computation Distance Computation

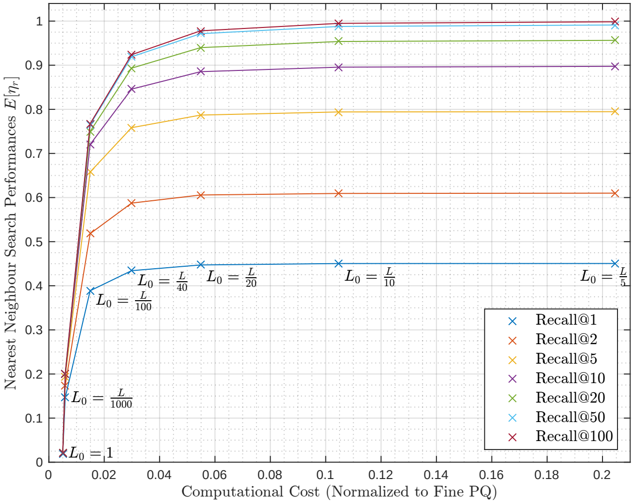
TESTED APPLICATIONS

Object Retrieval	Classification
Pictures querying within a large data collection, assuming the shape of SIFTs, i.e. extracted features local descriptors.	Identify labels associated to a large set of pictures containing handwritten digits, ranging from 0 to 9.
TEXMEX Group - INRIA, Rennes.	**MNIST Database - Yann LeCun**

In both cases, **all of the parameters of interest are trained** to work at the best settings compromise.

The achieved results are shown in the next couple of slides.

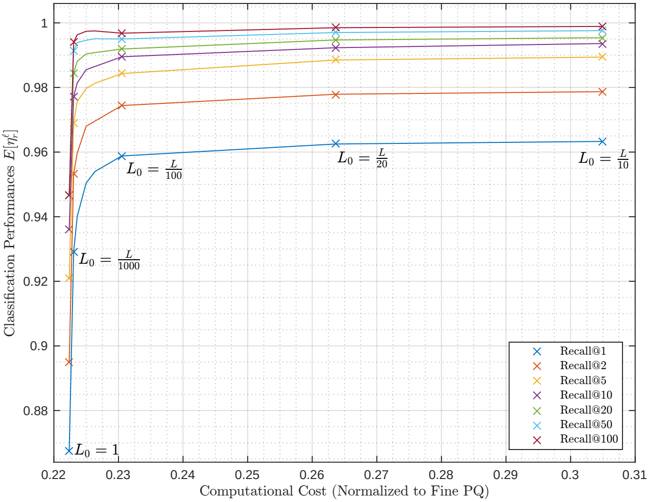
PERFORMANCES: OBJECT RETRIEVAL



Parameters list:

- Dimensionality $d = 128$
- Cardinality $N = 1000000$
- Coarse PQ $k_c = 2, m_c = 32$
- Fine PQ $k_f = 16, m_f = 256$
- Number of networks $L = 10000$

PERFORMANCES: CLASSIFICATION



Parameters list:

- Dimensionality $d = 784$
- Cardinality $N = 60000$
- Coarse PQ $k_c = 2, m_c = 32$
- Fine PQ $k_f = 16, m_f = 256$
- Number of networks $L = 2000$

CONCLUSIONS

CONCLUSIONS

This work illustrates the interest of using the introduced neural networks model to accelerate ANN search over real applications.

Future works may include different new proposals, such as:

→ Getting rid of the **Fine search** stage by implementing message retrieval strategies based with a Bayesian approach;

→ Trying to use a **Hierarchical approach**, consisting in cascading different granularity layers;

→ Proposing **refined allocation strategies** to choose which vector should be stored in which networks;

e.g. Group testing, hence the selection of non-disjoint sets

Thank you for your kind attention.

BIBLIOGRAPHY

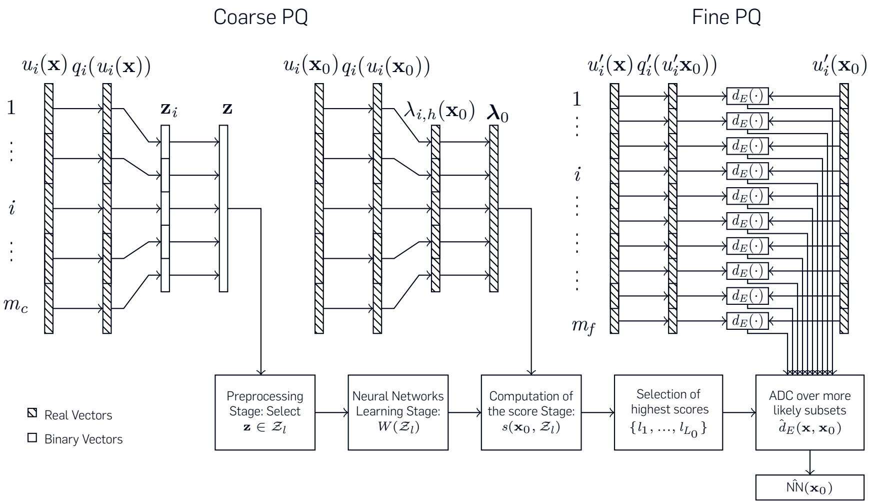
Demetrio Ferro, Vincent Gripon, and Xiaoran Jiang.
Nearest neighbour search using binary neuroinspired associative memories.
In Neural Information Processing Systems, November 2015.
Submitted.

Demetrio Ferro, Vincent Gripon, and Xiaoran Jiang.
Nearest neighbour search using binary neuroinspired associative memories.
In Computer Vision and Pattern Recognition, November 2015.
Submitted.

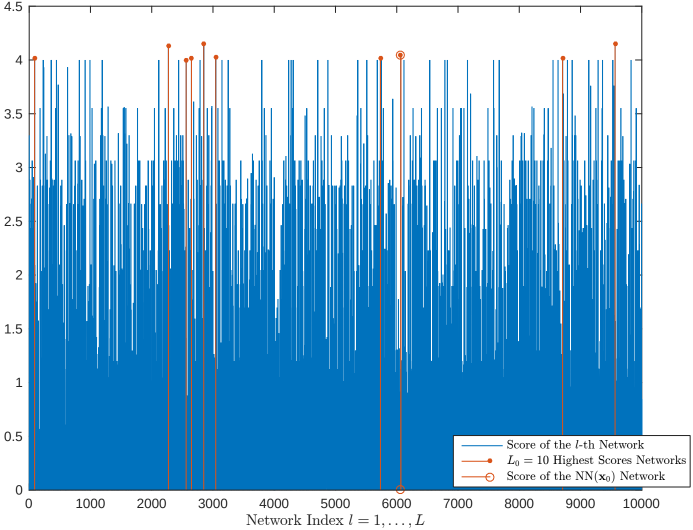
Hervé Jégou, Matthijs Douze, and Cordelia Schmid.
Product Quantization for Nearest Neighbour Search.
IEEE Transac. on Pattern Analysis and Machine Intelligence, January 2011.

Chendi Yu, Vincent Gripon, Xiaoran Jiang, and Hervé Jégou.
Neural associative memories as accelerators for binary vector search.
In Proceedings of Cognitive, March 2015.
To appear.

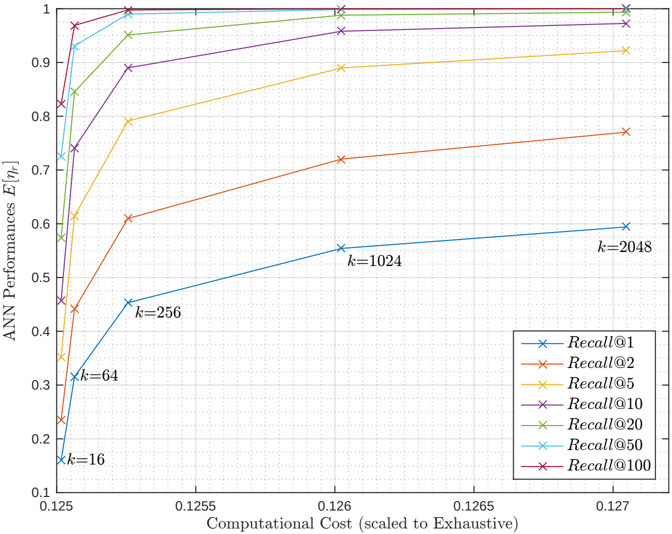
BACKUP - OVERALL TECHNIQUE SCHEME



BACKUP - RELIABILITY OF THE SCORES



BACKUP - SELECTING PARAMETERS: PQ PERFORMANCES



BACKUP - SELECTING PARAMETERS: CAOARSE GRANULARITY