

CM1K, KNN in μS , any Dataset Size

Technical Brief

The k -nearest neighbor algorithm (k -NN) is a method for classifying objects based on closest training examples in the feature space. This paper explains how the parallel architecture of the CogniMem chip makes it the fastest candidate to retrieve the K closest neighbors of a vector among ANY number by (1) calculating distances in parallel and (2) sorting them in increasing order autonomously.

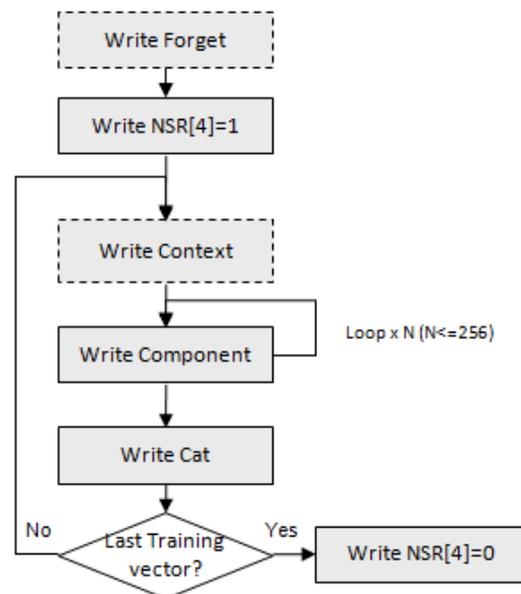
Loading the Training Examples

The training examples can be any number of vectors composed of up to 256 bytes. A vector can be a series of measurements with different dimensions and characteristics of a population of objects. In some cases, the 256 bytes can be samples of the temporal evolution of a signal, or the spatial distribution of pixels, and more.

The training examples are loaded sequentially into the neurons using the Save and Restore (SR) mode of the CM1K chip. In this mode, the neurons are passive and writing a neuron register takes one system clock cycle. The following diagram describes the simple sequence of commands to load the training examples in the neurons. Its execution time is solely proportional to the number of examples and independent of the architecture of the neural network whether it is composed of a single CM1K chip (1024 neurons) or a chain of 10, 100 or more CM1K chips daisy chained together.

The initial Write Forget resets the category of all the neurons to zero. If you do not execute this command, you will be appending the examples to the ones already loaded in the existing committed neurons.

The neurons are then set to the Save and Restore mode by setting bit 4 of the Neuron Status Register (NSR) to 1 and the first neuron of the chain becomes "Ready-To-Load". The N bytes of the first example are loaded through a series of Write Components. After the N^{th} component, the Write Category assigns a value to the Category register of the neuron (default is 1). The latter becomes "Committed" and the next neuron in the chain becomes "Ready-To-Load". Once all examples have been loaded, the network is returned to its default Learn and Recognize mode by setting bit 4 of the NSR back to zero.



Depending on the execution, or not, of the two optional commands shown in dotted frames, the loading of the training examples will take

(N+1) or (N+2) cycles per vector plus an overhead of 2 or 3 cycles.

Smart Category Assignment

Whenever possible, a good use of the category register consists of assigning a different category value to each training example. This will ensure if the recognition of a vector causes two or more neurons to report the same distance, their response will be differentiated from one another by their category value (refer to the paragraph "Limitations of the CM1K chip" later in this paper).

If the training examples have pre-defined categories which can be encoded on less than 15 bits, it is a good idea to edit the unused bits to include some indexing into the neuron's category register. Again, this will minimize the probability that later multiple neurons fire with the same distance and same category and be counted as one neighbor.

Optional Context Usage

Writing the Context register is optional and only useful if you wish to load at once several sets of training vectors which are not related. The vectors can, for example, derive from different data sources or the same data source but with different feature extraction techniques. Under these circumstances the context can be an index to a table describing how the vector was obtained. Another usage of the context can be to encode the length of the vector, a time stamp, or otherwise.

At the time of recognition, the context of the input vector has to be assigned to the proper value so only the neurons belonging to the same context participate in the recognition.

Learning, not Loading, the Examples

It is possible to load the neurons with training vectors using the default Learn and Recognize mode. This method will take longer with N+19 clock cycles per vector instead of N+1 and create a decision space where zones of unknown and uncertainties can exist. Nevertheless, these zones will be discarded when it is time to recognize a vector with the KNN classifier since in KNN all the neurons fire whatever their influence field is.

When loading examples in Save and Restore mode, the number of committed neurons is equal to the number of examples. Furthermore since the neurons are passive and not reactive, erroneous inputs (if any) will not be detected, including duplicate examples, inconsistent examples, etc.

When learning examples in the Learn and Recognition mode, the neurons are reactive and only retain examples which bring novelty at the time they are learned. This means that the committed neurons are directly related to the order of the examples. This dependency can be waived by learning several times until no new neuron gets committed between two passes. Other factors affecting the learning behavior of the neurons are the Maximum and Minimum Influence Fields of the network. These parameters are global registers and help moderate the conservatism of the neurons.

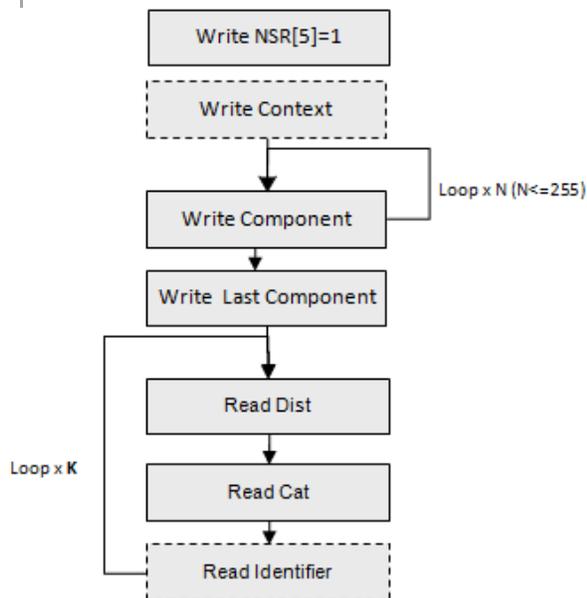
Closest K Neighbors in Microseconds

Broadcast the Vector to all Neurons

A new vector can be broadcasted to all of the neurons in parallel through a series of Write Component and a Write Last Component commands. After each input component, the neurons update their distance registers automatically according to the norm defined by bit 7 of their context register. Upon receipt of the last component their distance register gives the distance between the input vector and the reference pattern they hold in memory.

If an input vector is composed of 200 bytes and broadcasted to a single CM1K chip, the 1024 distances of the 1024 neurons in the chip are available after $200 + 2$ clock cycles. Similarly, if the network is a chain of 10 CM1K chips or 10,240 neurons, the 10,240 distance values are also available after the $200 + 2$ clock cycles.

Search and Sort in a Fixed Amount of Time



Once a new vector is broadcasted to all the neurons, the response of the top K neurons with the best matches is read out through K successive Read Distance and Read Category steps.

The autonomous sorting mechanism of the neurons is a key feature pertaining to their parallel architecture and a patented Search and Sort algorithm which allows each neuron to know if other neurons have a smaller distance value without the need for a supervisor or controller. In such cases, the neuron holds its response letting the one with a smaller distance respond first. The value K must be less than or equal to the number of committed neurons in the network.

Setting bit 5 of the Network Status Register to 1 switches the behavior of the neurons from the default Radial Basis Function classifier to the K-Nearest Neighbor classifier.

The initial Write Context command is optional as described in the previous chapter. The Read Category command is mandatory even if its value has no particular interest. If it is not executed the neuron will remain in the race for the next Search and Sort thus preventing the retrieval of the following best matches. The Read Identifier can be optional and returns the index of the firing neuron. This index is assigned internally by the neurons at the time they get committed. Reading the identifier is useless if the category value has been taught as the example's number as suggested earlier in this document.

Same Distance, Multiple Categories

The response of the firing neurons is ordered per increasing distance and for the same distance per increasing category. The CM1K chip does not provide for the subsequent sorting per identifier.

If one or more of these neurons have the same distance and same category, the readout of the Distance register followed by the Category register will exclude them all at once from the next search and sort. If you are interested in surveying the histogram of the distances and a probability density function, this means that the neurons with the same distance and same category will be counted as one and produce incorrect results. That is why it is highly recommended to encode a different category value for each training example.

CM1K Timing Benchmarks

The number of clock cycles to access the CM1K registers which are useful for a KNN classification are listed below:

Register	Read	Write
Context	n/a	1
Component	1	1
Last Component	n/a	3
Distance	18	n/a
Category	3 if identified 19 otherwise	1 if identified 19 otherwise
Identifier	1	

Step 1: Broadcast a Vector with N Dimensions

Shortest Timing: $N+2$ (without changing the Context register)

Longest Timing: $N+3$ (with change of the Context register)

Step 2: Readout of the K Nearest Neighbors

Shortest Timing: $K * 22$ (all neurons loaded with the same category, request to read the identifier)

Longest Timing: $K * 39$ (all neurons loaded with a different category, request to read the identifier)

Benchmarks

In the following example, the CM1K chip is running at 27 MHz or 37 nanoseconds per clock cycle.

The values N and K are set respectively to 96 and 20 as a reference to a benchmark published for the CUDA architecture from NVIDIA.

The broadcast of a single vector of 96 values takes 3.63 microseconds. The "shortest" readout of the 20 nearest neighbors of a single vector takes 16.30 microseconds

Number of input vectors	CogniMem NN
9600	156.44
19200	312.89
38400	625.78