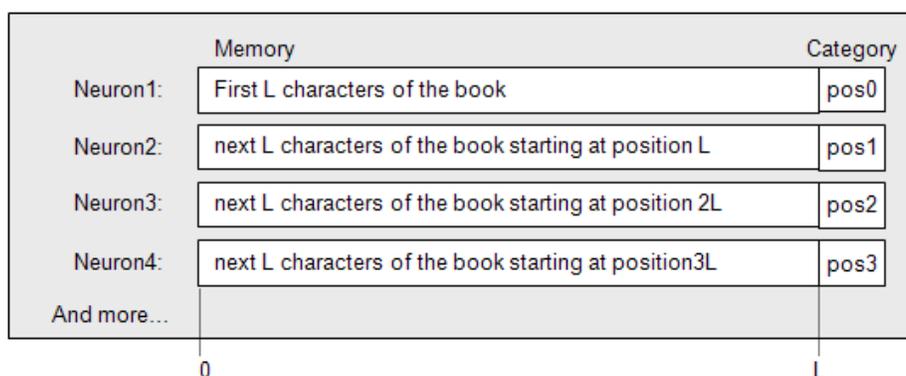# Searching a String in a Data Base – Method & Speed Performance

The parallel architecture of the CogniMem neural network makes it possible to find in a text body a word, or combination of words, in a number of clock cycles independent from the length of the body text.

Usage is text recognition, DNA sequence finding, bioinformatics, internet address matching, etc.

Step 1: Learn a Body Text

1) The body text of total length M is segmented into strings of L characters, with optional cropping and padding to avoid a segment containing a split word.
   a. The neurons of the CM1K have a memory of 256 bytes, so L must be ≤ 256.
   b. The CM1K chip features 1024 neurons, so a single CM1K chip can store up to 262,144 characters.
   c. Multiple CM1K chips can be cascaded to extend the total capacity of the network per increment of 1024 neurons. Text composed of 1 million characters can be stored in 977 CM1K chips in parallel.

2) The segments are loaded in the neurons using the CM1K chip in Save and Restore mode along with a category equal to the number of segments in the body text, or the position of the first character of the segment in the body text.

   a. Loading a segment of L characters into the neurons takes only L+1 clock cycle in Save and Restore mode.
   b. The loading time can be optimized by designing an architecture where a chain of CM1K chips can be loaded in parallel.
   c. The category register is coded on 15 bits and can have a value between 0 and 32767. If the body text is composed of more than 32767 segments, the usage of the context register of the CM1K will allow entering 127 times more segments. In this case the position of the segment in the body text will be decoded by reading its context register and category register.
   d. Changing the context will take one extra clock cycle every 32767 segments.

	Memory	Category
Neuron1:	First L characters of the book	pos0
Neuron2:	next L characters of the book starting at position L	pos1
Neuron3:	next L characters of the book starting at position 2L	pos2
Neuron4:	next L characters of the book starting at position3L	pos3
And more…		

0   L

CogniMem Technologies, Inc.
81 Blue Ravine Road, Suite 240, Folsom, CA 95630
916.358.9483 F 916.358.9485

www.cognimem.com
March 2010
P a g e | 1

Register Transfer Level Loading Sequence	Cycles
Write NSR to set the network in Save/Restore mode	1
Write RESETCHAIN	1
Loop (M÷L) times	(M÷L)*(
Write (L-1) COMP	(L-1)
Write LCOMP	3
Write CAT	1
End loop	)
Write NSR to set the network to Learn/Recognition mode	1
Total Cycles =	(M÷L)*(L+3) + 3
Body text of 1 million characters M=1,000,000, L=256	1,004,000 cycles
@27Mhz clock	37 seconds

Step 2: Find a Word in the Text

The recognition of a word of length W is made by sliding it over the (L-W) possible positions (P) in a segment (i.e. neuron memory). At each position P, the W characters are broadcasted to all the neurons and the response of the firing neurons, if any, can be read per increment of 36 clock cycles to read their category and distance registers.
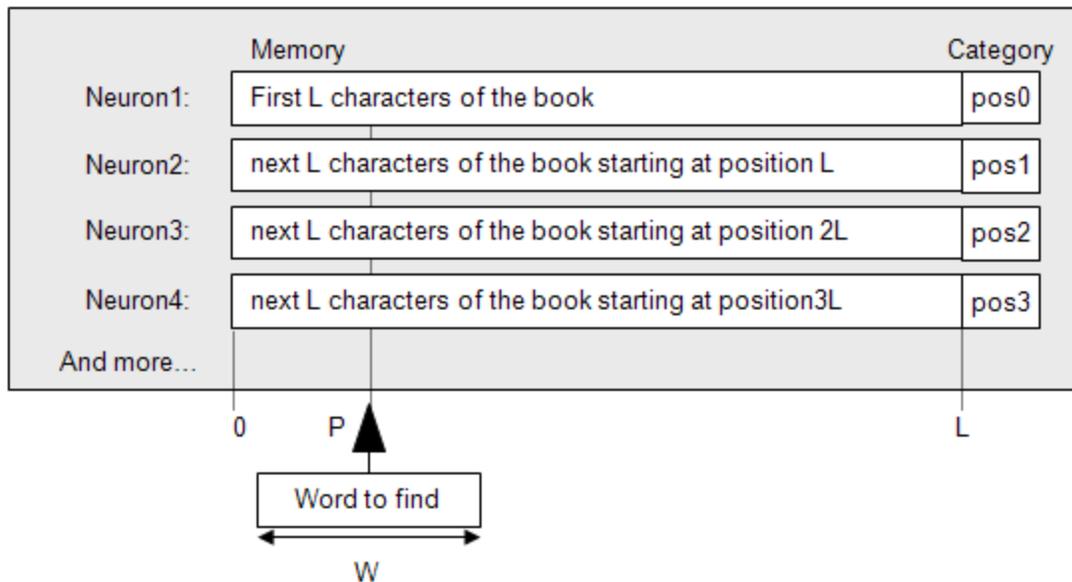
Remark #1

An important feature of the CM1K for this application is the ability to disable the distance evaluation unit of the neurons while moving in their memory cells. This allows starting the recognition at a given position P and also discarding some wild characters (*, ?, etc.) inserted in the word to recognize. This feature is the INDEXCOMP register.

CogniMem Technologies, Inc.
81 Blue Ravine Road, Suite 240, Folsom, CA 95630
916.358.9483 F 916.358.9485

www.cognimem.com
March 2010
P a g e | 2

Remark #2

The recognition time of a word does not depend on the length of the body text, but on the number of positions of the word in the neuron's memory and the number of firing neurons per position.

Remark #3

The response of the neurons is read per increasing order of distance and not position. This distance refers to the difference between the ASCII codes of the word to recognize and the W characters in the neuron memory at position P.
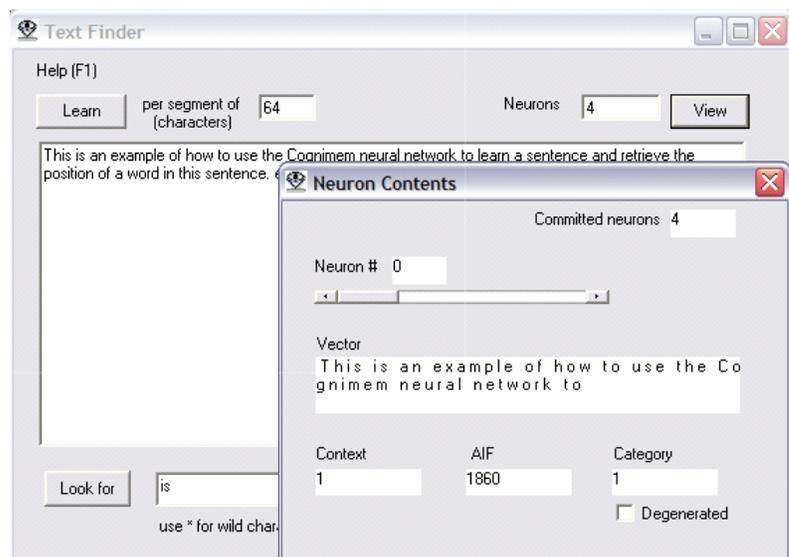
Register Transfer Level Recognition Sequence	Cycles
Loop from P=0 to L-W	(L-W) *(
Move to the memory cell P in all the neurons at once	
Write INDEXCOMP	1
Broadcast the word of length W	
Write (W-1) COMP	W-1
Write LCOMP	3
Read results of all k firing neurons	k *(
Read DIST	18
Read CAT	19
End of Loop	)
End of Loop	)
Total Cycles =	(L-W) * (W+3+(k*37))
W=8, L=256, k=0	2,728 cycles
W=8, L=256, k=1 (assuming one firing for each position P)	11,904 cycles
@27Mhz clock	101µsec
	440µsec

Examples of Code

The code provided below is an extract of a Text Recognition example developed by CogniMem Technologies. It is written in Visual Basic, but easy to understand and convert to other programming languages.

The Text Recognition example displays the following sentence as default:

 "This is an example of how to use the CogniMem neural network to learn a sentence, retrieve the position of a word in this sentence, and edit a new sentence to look for a word of your choice."

CogniMem Technologies, Inc.
81 Blue Ravine Road, Suite 240, Folsom, CA 95630
916.358.9483 F 916.358.9485

www.cognimem.com
March 2010
P a g e | 3

You can build different knowledge bases by changing the number of characters per segment and clicking Learn. You can then review the portion of text held by each neuron. The category is the position of the first character in the input text.

Knowledge 1 with L=256

Creates a single neuron containing the entire sentence "This is an example of how to use the CogniMem neural network to learn a sentence, retrieve the position of a word in this sentence, and edit a new sentence to look for a word of your choice."

Knowledge 2 with L=64

Neuron1 contains "This is an example of how to use the CogniMem neural network to"
Neuron2 contains "learn a sentence, retrieve the position of a word in this"
Neuron3 contains "sentence, and edit a new sentence to look for a word of your"
Neuron4 contains "choice"

1) *Learning the Text of Reference*

```
   'Clear the neurons before building a new knowledge base
   CM_Init
   TextRemaining = TextToLearn
   Pos = 1
   'switch to the save and restore mode
   CM_Write CM_NSR, 16
   CM_Write CM_RESETCHAIN, 0
   While Len(TextRemaining) <> 0
   'find position of the last characters of the last word in the next segment
       If Len(TextRemaining) <= L Then TempLength = Len(TextRemaining)
       Else
           For j = L To 1 Step -1
               If Mid(TextRemaining, j, 1) = " " Then Exit For
           Next j
           TempLength = j
       End If
   'fill the vector and pad with spaces if needed
       For j = 0 To TempLength – 1
       Vector(j) = Asc(Mid(TextRemaining, j + 1, 1))
       Next j
       For j = TempLength To L - 1: Vector(j) = 0: Next j
   'learn the vector and assign a category equal to the position of the first character in the
sentence
       For j = 0 To L - 1: CM_Write CM_COMP, Vector(j): Next j
       CM_Write CM_CAT, Pos
       Pos = Pos + TempLength
       TextRemaining = Mid(TextRemaining, TempLength + 1, Len(TextRemaining))
   Wend
   'cancel the save and restore mode
   CM_Write CM_NSR, 16
```

2) *Finding a Word in the Text of Reference*

```
   Hit = 0: TxtHit = ""
   For Pos1 = 0 To L - WordLength - 1

   'At each new position, reset the distance registers of all neurons without 'erasing their
memory. Select the KNN classifier at the same time
       CM_Write CM_NSR, 32

   'point to the component Pos1 and broadcast the characters of the word to the 'neurons. If
this character is the wild character, simply skip to the neuron 'next component without updating
the distance register.
```

CogniMem Technologies, Inc.   www.cognimem.com
81 Blue Ravine Road, Suite 240, Folsom, CA 95630   March 2010
916.358.9483 F 916.358.9485   P a g e | 4

```
CM_Write CM_INDEXCOMP, Pos1
For j = 0 To WordLength - 2:
    'Test for the wild component (code 42)
        If VectorToReco(j) <> 42 Then CM_Write CM_COMP, VectorToReco(j) Else CM_Write
    CM_INDEXCOMP, Pos1+j+1
Next j
CM_Write CM_LCOMP, VectorToReco(WordLength - 1)

'look for the exact matches and convert the category into an absolute 'position in the
sentence.
If CM_Read(CM_DIST) = 0 Then
    Do
        Pos2 = CM_Read(CM_CAT) And &H7FFF
        If Pos2 = &H7FFF Then Exit Do
        Matches(Hit) = Pos1 + Pos2 - 1
        Hit = Hit + 1
    Loop
End If

Next Pos1
```

Note:

The matches are listed in order of occurrence with respect to Pos1 which is not the position within the input sentence but the relative position within the neuron memories.

CogniMem Technologies, Inc.   www.cognimem.com
81 Blue Ravine Road, Suite 240, Folsom, CA 95630   March 2010
916.358.9483 F 916.358.9485   P a g e | 5