ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/3566454

CNN based on multi-valued neuron as a model of associative memory for grey scale images

Conference Paper · November 1992

DOI: 10.1109/CNNA.1992.274330 · Source: IEEE Xplore

CITATIONS
81

READS
14

2 authors, including:

Igor Aizenberg
Texas A&M University - Texarkana

103 PUBLICATIONS **1,081** CITATIONS

SEE PROFILE

Available from: Igor Aizenberg
Retrieved on: 24 May 2016

CNN BASED ON MULTI-VALUED NEURON AS A MODEL OF ASSOCIATIVE MEMORY FOR GREY-SCALE IMAGES

Naum N. Aizenberg+, Igor N. Aizenberg++

+ University of Uzhgorod, professor of the Department of Cybernetics; Geroev Stalingrada 28, kv.49, Uzhgorod, 249015, Ukraine
Phone: (+7 03122) 23908, Fax: (+7 03122) 36120

++ Joint Venture PGD, Research Department Chief, Dr.Sc.; Engelsa 27, kv.32, Uzhgorod, 294015, Ukraine
Phone: (+7 03122) 63269, Fax: (+7 03122) 36120

This work is supported by Maltian company DKL Ltd - 198 Old Bakery Street, Valetta, Malta. Fax: (+356) 221893

Abstract

In this paper we consider mathematical model of multi-valued neural element which is suggested to be used as one of the alternatives of the basic CNN element. On the basis of such CNN we offer the model of the associative memory for storing grey-scale images. Efficient learning algorithm for neural element and associative memory is presented in the paper.

I. INTRODUCTION

CNN, introduced in [1] and having been extensively developed recently [2 and others], became brilliant alternative to conventional computers for image processing and recognition.

Specifically, on the basis of CNN were introduced a number of models of associative memory and were developed the methods of solving the problems of image recognition and filtering [2 - 4]. Despite the fact that a lot of complicated tasks are solved on the basis of CNN, still we assume that in a great number of papers the range of processed signals is "unfairly" restricted - either mostly binary (bipolar) or exclusively binary signals (in case of associative memory designing) are considered. To break this restriction and extend both CNN functionality and the range of problems that are solved on CNN basis, we suggest using neural element based on the mathematical model of multi-valued threshold element [5] as a basic CNN neuron, as well as an alternative to neurons considered in [1, 6]. Proceeding from the idea that application of this element is natural for multi-valued signals processing and

0-7803-0875-1/92 3$.00 ©1992IEEE

Authorized licensed use limited to: Texas A M University. Downloaded on September 2, 2009 at 18:35 from IEEE Xplore. Restrictions apply.

considering the quickly convergent learning algorithm (presented below), we believe that application of this neuron can be highly efficient. For instance, designing cellular associative memory for storing grey-scale images and implementation of image filtering algorithms on such CNN is absolutely natural. Another advantage of this basic CNN neuron is convenience of its analog (including optical) hardware implementation, since operations with complex numbers provide "arithmetical basis" of the neuron.

II. MULTI-VALUED CNN BASIC ELEMENT

To simplify the analysis, we will consider discrete time version of the CNN, though it is evident that all the results presented below are easily trasformed to the continuous time case.

Suppose we have CNN of a dimension $N \cdot M$. We will apply ***multi-valued neural elements*** as basic neurons of this network. Each of these elements performs the following transformation:

$$Y_{ij}(t+1) = F\left[\, W_0 + \sum_m W_m^{ij} X_m^{ij}(t) \,\right], \qquad (1)$$

where Y_{ij} - neuron state, W_m^{ij}, X_m^{ij}- connection weights corresponding to m-th input of ij-th neuron and input signal value on m-th input of ij-th neuron respectively. $F(\cdot)$ - output function of ij-th neuron that will be defined furtheron.

Let input signals X and output signal Y for each neuron be located in the range $0, \dots, k-1$, i.e. each neuron at each particular moment performs some function of k-valued logic determined by weights W. It is evident that for byte images $k = 256$.

Let's assume that each value of $0, \dots, k-1$ corresponds to complex number r_j, $j = 0, 1, \dots, k-1$ like this:

$$r_j = exp(i \cdot 2\pi \cdot j/k) \qquad (2)$$

where i is an imagenary unit. E.g., $r_0 = 1$, $r_1 = \varepsilon$, primitive k-th power root of a unit, $r_2 = \varepsilon^2, \dots, r_{k-1} = \varepsilon^{k-1}$. Following this input and output signals for each neuron will be coded by k-th power roots of a unit.

Let $arg(Z)$ - argument of complex number Z ($0 \leqslant arg(Z) < 2\pi$). Now we can determine output function F for neuron (refer to (1)). Let's define function $CSIGN(Z)$, $Z \subset \mathbf{C}$ (C-Complex Numbers Field) as follows:

Authorized licensed use limited to: Texas A M University. Downloaded on September 2, 2009 at 18:35 from IEEE Xplore. Restrictions apply.

$$\mathbf{CSIGN}(z) = exp(i \cdot 2\pi \cdot j/k) = \varepsilon, \qquad \text{if } 2\pi \cdot j/k \leqslant arg(Z) < 2\pi(j+1)/k,$$
$$\mathbf{CSIGN}(0) = \varepsilon^0 = 1. \qquad (3)$$

The interpretation of the (3) may be the next. If complex plane is separated into k equal sectors and complex number Z is located in j-th sector then function CSIGN(Z) equals ε^j (j = 0, 1, ..., k-1; ε is a primitive k-th power root of a unit).

Specifically CSIGN function will be used as output function F of the neuron. In this case (1) will be transformed to:

$$Y_{ij}(t+1) = \mathbf{CSIGN} \left[W_0 + \sum_m W_m^{ij} X_m^{ij}(t) \right]. \qquad (4)$$

So, (4) describes dynamics of the ij-th neuron. Naturally, since Y and X are complex numbers then weights W are complex too. In specific case when k = 2 and weights are real, (4) will describe linear threshold element and according to (3) CSIGN will become natural function SIGN.

III. LEARNING ALGORITHM FOR MULTI-VALUED NEURON

Further we will describe learning algorithm for multi-valued neural element. In the case k = 2 the problem is reduced to the well-known problem of Rosenblatt's Perceptron learning. Here we will consider the case when $k \geqslant 2$, paying special attention to the case $k > 2$.

Let $k \geqslant 2$ be an arbitrary natural number. Let's have k of non-intersecting learning subsets $A_j = \{X_1^j, ..., X_N^j\}$; j = 0, 1, ..., k-1, $X = (X_0, X_1, ..., X_n)$, where $X_0 = 1$, and other coordinates take their values from the set $\{\varepsilon^0, \varepsilon^1, ..., \varepsilon^{k-1}\}$. When $A_0, \ldots, A_{k-1}$ are predetermined, the task of learning consists in figuring out of permutation $(\alpha_0, \alpha_1, ..., \alpha_{k-1})$ and vector $W = (W_0, W_1, ..., W_n)$ satisfying

$$\mathrm{CSIGN}(X, \bar{W}) = \varepsilon^{\alpha_j} \qquad (5)$$

for each $X \in A_j$, j = 0, 1, ..., k-1, where $\bar{W} = (\bar{w}_0, \bar{w}_1, ..., \bar{w}_n)$ - vector complex-conjugated to W; $(X,\bar{W}) = w_0 + w_1x_1 + ... + w_nx_n$ - scalar product of (n+1)-dimensional complex-valued vectors in (n+1)-dimensional space. If (5) is true, sets $A_0, A_1, ..., A_{k-1}$ are called *edge-separable*.

Since we consider neural network, we can assume that permutation $(\alpha_0, \alpha_1,$

Authorized licensed use limited to: Texas A M University. Downloaded on September 2, 2009 at 18:35 from IEEE Xplore. Restrictions apply.

..., α_{k-1}) is known, because it will be determined by the state of neurons from nearest neighbourhood of given neuron.

Now we have to make iterative evaluation of sequence S_w of weight vectors $W_0, W_1, \ldots$, for it to satisfy condition (5) starting from some number m and $W_m = W_{m+1} = \ldots$.

The sequence S_w is evaluated as follows:

$$W_{m+1} = W_m + \omega_{s_m} \cdot D_m \cdot \varepsilon^{\alpha_j} \cdot \bar{X}_m , \qquad (6)$$

where $D_m > 0$ - correction coefficient, s_m is equal 0, 1, 2 or 3 depending on one of the following cases:

a). $\omega_0 = 0$, if $\quad \mathrm{CSIGN}(X_m, \bar{W}_m) = \varepsilon^{\alpha_j}$;

b). $\omega_1 = -i\varepsilon$, if $\quad \mathrm{CSIGN}(X_m, \bar{W}_m) = \varepsilon^{\alpha_j+1}$ for $k = 2, 3$ and

$$\varepsilon^{\alpha_j+1} \prec \mathrm{CSIGN}(X_m, \bar{W}_m) \prec \varepsilon^{\alpha_j+[k/4]} \quad \text{for } k \geqslant 4;$$

c). $\omega_2 = 1$, if

$$\varepsilon^{\alpha_j+[k/4]+1} \prec \mathrm{CSIGN}(X_m, \bar{W}_m) \prec \varepsilon^{\alpha_j+3[k/4]-1} \quad \text{for } k \geqslant 4;$$

d). $\omega_3 = i$, if

$$\mathrm{CSIGN}(X_m, \bar{W}_m) = \varepsilon^{\alpha_j+2} \quad \text{for } k = 3 \text{ and}$$

$$\varepsilon^{\alpha_j+3[k/4]} \prec \mathrm{CSIGN}(X_m, \bar{W}_m) \prec \varepsilon^{\alpha_j+k-1} \quad \text{for } k \geqslant 4.$$

Here $[k/4]$ is an integer part of $k/4$, " $\prec$ " is defined as follows:

$$\varepsilon^p \prec \varepsilon^q \iff p \leqslant q \pmod k.$$

Let's clarify the meaning of rule (6). Note that in our case $D_m = D = 1$, $m = 0, 1, \ldots$, because all the components of all input vectors X have a constant modulo (magnitude) equal to 1. In case a) vector $W = W_m$ satisfies (5) when $X = X_m$, so there is no reason to change it. In cases b) - d) weight vector is corrected by component-by-component addition to W_m of some vector depending in each case of the difference between $\varepsilon^{-\alpha_j}\mathrm{CSIGN}(X_m, \bar{W}_m)$ and ε^0. The second

Authorized licensed use limited to: Texas A M University. Downloaded on September 2, 2009 at 18:35 from IEEE Xplore. Restrictions apply.

addendum in (6) is evaluated from following reason:

$$(X_m, \bar{W}_{m+1}) = (X_m, \bar{W}_m) + (X_m, \omega_{s_m} \cdot \varepsilon^{\alpha_j} \cdot X_m) =$$

$$= (X_m, \bar{W}_m) + \omega_{s_m} \cdot \varepsilon^{\alpha_j}(X_m, X_m) = (X_m, \bar{W}_m) + (n+1)\omega_{s_m} \cdot \varepsilon^{\alpha_j},$$

where n is number of neuron inputs. ω_{s_m} is introduced to make the value of expression

$$\mathrm{CSIGN}(X_m, \bar{W}_{m+1}) = \mathrm{CSIGN}\,[(X_m, \bar{W}_m) + (n+1)\omega_{s_m} \cdot \varepsilon^{\alpha_j}]$$

closer to ε^{α_j} at each next iteration.

Initial value W_0 for vector W can be arbitrary (furtheron we will consider its selection for associative memory learning).

So iterative learning process is reduced to sequentional application of formula (6) taking into account rules a) - d) till case a) is reached. It is evident that if α_j is fixed, this process is converged maximum on step 5-6. If permutation $(\alpha_0, \alpha_1, \ldots, \alpha_{k-1})$ is fixed, then $W_0^{\alpha_0}$ is the first to be evaluated according to (6). Then we use the obtained vector for evaluation of $W_0^{\alpha_1}$, etc. , every time checking new weight vector for previous α_j. This procedure goes on until vector W_m satisfies (5).

THEOREM I. (On convergence of learning algorithm). If learning subsets A_0, A_1, ..., A_{k-1} are edge-separable, then it will take finite number of steps to obtain vector W_m satisfying (5) for all learning subsets.

If we presume that learning process is infinite then we will obtain contradiction with edge-separation of learning subsets.

So investigation of the possibility of neuron learning in each particular case is reduced to the investigation of the edge-separation of learning subsets. Note if this issue is settled positevely, iterative learning process is converged pretty fast. The number of iterations can reach hundreds, sometimes - thousands, but not more.

IV. MODEL OF ASSOCIATIVE MEMORY ON THE BASE OF CNN FROM MULTI-VALUED NEURONS

For solving the task of storing m patterns (grey-scale images) determined by matrixes P^1, P^2, ..., P^m of $N * M$ dimension and each image contains k of

Authorized licensed use limited to: Texas A M University. Downloaded on September 2, 2009 at 18:35 from IEEE Xplore. Restrictions apply.

grey-scales: $0, 1, \ldots, k\text{-}1$, let's use CNN based on multi-valued neurons. The task is reduced to evaluation of such weights W that any of P^i patterns could be reconstructed during finite number of steps from the input pattern $\tilde{P}^i$ that differs from P^i in a number of points.

So let's recode given patterns P from integer-valued to complex-valued $(0 \longrightarrow \varepsilon^0, \ldots, (k\text{-}1) \longrightarrow \varepsilon^{(k-1)})$. Then for evaluation of the connected weights between ij-th and pq-th neurons we will use the following Hebb rule [3] generalization for complex-valued patterns:

$$W^0_{ij,pq} = \begin{cases} 0 & \text{for } |i\text{-}p| > 1 \text{ or } |j\text{-}q| > 1 \\ \sum_{l=1}^{m} \bar{P}^{\,l}_{ij} P^l_{pq} & \text{for } |i\text{-}p| \leqslant 1 \text{ or } |j\text{-}q| \leqslant 1 \end{cases}$$

where P^l_{ij} is an ij-th point of l-th input pattern, " $\bar{\ }$ " is the complex-conjugation.

Having evaluated all W^0 weights for all the neurons, we obtain initial value for weights of our associative memory. Further, for evaluation of final values of weights W that will determine dynamics of the associative memory, it is necessary to apply learning algorithm based on (6) for each neuron from CNN. It is evident that the speed of learning algorithm operation will be the higher the more patterns P are correlated among themselves or , which is the same in our case, the less is the Euclid distance between patterns.

References

[1] L.O. Chua and L. Yang, "Cellular neural networks: Theory", IEEE Trans. Circuits Syst., Vol. 35, pp. 1257-1290, Oct. 1988

[2] "Proceedings of the 1990 IEEE International Workshop on CNN and their applications (CNNA-90)", Budapest, 1990, IEEE Catalog No. 90TH0312-9.

[3] S.Tan, J.Hao and J.Vandevalle, "Cellular Neural Networks as a Model of Associative Memories", in the same Proceedings, pp. 26-35.

[4] G.G.Yang, "Optical Associative Memory with Invariances", in the same Proceedings, pp. 291-301.

[5] N.N.Aizenberg, Y.L.Ivaskiv, "Multi-valued Threshold Logic", Naukova Dumka Publisher House, Kiev, 1977 (in Russian).

[6] L.O.Chua, L.Yang, K.R.Krieg, "Signal Processing using Cellular Neural Networks", Journal of VLSI Signal Processing, Vol. 3, No. 1/2 1991, pp. 25-51.

Authorized licensed use limited to: Texas A M University. Downloaded on September 2, 2009 at 18:35 from IEEE Xplore. Restrictions apply.