A k-winners-take-all neural net

Bruce D. Calvert
email calvert@math.auckland.ac.nz
Department of Mathematics,
University of Auckland, Private Bag 92019,
Auckland, New Zealand.

Corneliu A. Marinov, e-mail gabinic@hades.ro
Department of Electrical Engineering,
Polytechnic University of Bucharest,
77206, Bucharest, Romania

Abstract

An analog Hopfield type neural network is given, that identifies the K largest components of a list d of N real numbers. The list to be processed is a summand of the input currents of the neurons, and the network is started from 0. We provide easily computable restrictions on the parameters. The trajectories are shown to eventually have positive components precisely in the positions given by the K largest elements in the input list.

1 Introduction

We are interested in the synthesis and rigorous analysis of a particular type of neural circuit, called a "K-winners-take-all" network, or $KWTA$ network for short, and in this presentation we summarise the results of [1]. Selecting the K largest elements from a data set of N numbers is a key task in decision making, pattern recognition, associative memories or competitive learning networks [2]–[6]. $KWTA$ networks find applications in telecommunications, particularly for controlling data packet switches [7]. In [10] one shows applications in $VLSI$ auditory and visual systems, while in [9] a $KWTA$ network as an analog decoder of error-correcting codes is proposed.

2 Background Material

We are given N real numbers, d_1 to $d_N, N > 1$, and wish to choose the K largest, $1 \leq K \leq N-1$. We take them all to be unequal and define the permutation σ by

$$d_{\sigma(1)} > d_{\sigma(2)} \ldots > d_{\sigma(N)}, \tag{1}$$

saying that the vector $d = (d_1, \ldots, d_N)$ is σ-ordered. The conventional goal is to build an analog circuit, which excited by d, has an output $v(t)$ which settles to a steady state $\overline{v}$ such that

$$\overline{v}_{\sigma(i)} > 0,\ i \in \overline{1,K}, \text{ and } \overline{v}_{\sigma(i)} < 0,\ i \in \overline{K+1,N}. \tag{2}$$

The inequality (2) expresses the "$KWTA$ property": the components $\overline{v}_{\sigma(1)}$ to $\overline{v}_{\sigma(N)}$ "win" the competition, and the fact that they are the only positive components of $\overline{v}$ shows that $d_{\sigma(1)}$ to $d_{\sigma(K)}$ are the K largest components of d.

Another requirement for our $KWTA$ network is a finite working time. That is, we should provide a condition to stop the process at a finite time t^* rather than wait for the (theoretically) unattainable equilibrium point to be reached, However there is no one time after which we can stop, rather t^* depends on the list d. In our network, we shall use a stopping condition

$$|v_i(t)| \geq 1 - \varepsilon$$

for all i and some time t, where ϵ will be a chosen parameter, to establish that if

$$v_{\sigma(i)}(t) > 0,\ i \in \overline{1,K}, \text{ and } v_{\sigma(i)}(t) < 0,\ i \in \overline{K+1,N}, \tag{3}$$

then $d_{\sigma(1)}$ to $d_{\sigma(K)}$ are the K largest components of d. This avoids using a fixed time t to approximate $\overline{v}$ by $v(t)$, since there is none which is independent of d for our network.

Finally, each time a list has been processed, the machine should be reset at the state zero, in wait for the next list. It is not sufficient to switch the input currents to zero, and we calculate how the gain can be set lower so that zero becomes a stable equilibrium.

Let us describe our machine. We propose an artificial neural network of Hopfield type, having N neurons (amplifiers). Each neuron has an input-output characteristic given by $v_i = g(\lambda u_i)$, where $g : \boldsymbol{R} \to (-1,1) = tanh$. Here λ is a positive parameter known as the gain, controlling the slope of the characteristic. It will play a crucial role here. Each neuron has the same input resistance R and capacitance C, and we denote by ℓ and s the positive quantities $1/R$ and $1/C$ respectively. The interconnection matrix T is taken to be

$$\begin{pmatrix} a & p & \cdots & p \\ \vdots & & & \\ p & & & a \end{pmatrix} \tag{4}$$

where all diagonal terms are equal and positive, and all nondiagonal ones are equal and negative, making T symmetric. The network input currents $b_i, i \in \overline{1,N}$, additively include the quantities d_i to be processed; for each i,

$$b_i = d_i - 2pK + pN. \tag{5}$$

The neural network is described by the following differential equation with zero initial state.

$$\begin{cases} \dot{u} &= s[-\ell u + TG(\lambda u) + b]. \\ u(0) &= 0, \end{cases} \tag{6}$$

Here $G : \boldsymbol{R}^N \rightarrow (-1,1)^N$ is given by $G(x) = (g(x_1), \ldots g(x_n))$. For each λ, the function $t \mapsto u^\lambda(t)$, giving a solution of (7), is called a "u-solution". We may consider this dynamical system in terms of the variable $v = G(\lambda u)$. For each λ, the function $t \mapsto v^\lambda(t) = G(\lambda u^\lambda(t))$ is called a "v-solution".

The neural network model (7) or (9) has the following basic properties.

THEOREM 1 *(i) For each* $\lambda > 0, (7)$ *has a unique* C^1 *solution* $u^\lambda : [0,\infty) \rightarrow \boldsymbol{R}^N$.
(ii) $u^\lambda(t)$ *is bounded uniformly in* λ *and* t.
(iii) For each $t > 0$, $\lambda \mapsto u^\lambda(t)$ *is continuous on* $(0,\infty)$.
(iv) For each λ, *there is a u-equilibrium* $\overline{u}^\lambda$ *such that* $\lim\limits_{t\rightarrow\infty} u^\lambda(t) = \overline{u}^\lambda$.

These facts are valid for more general dynamical systems. In particular (iv) is valid for any system (7) where T is symmetric and g is of "sigmoidal type". In this case (7) is a gradient-like system. Finally, let us mention that the above results hold for v-solutions and v-equilibria as well.

3 Main Results.

Let us concentrate for now on the v-solution, with trajectories in the open hypercube $(-1,1)^N$. It starts at zero, and, with parameters correctly chosen, it finishes with components split into positive and negative in accordance with (3). We refer to the 2^N points of $\{-1,1\}^N$ as corners. Among the corners there exists only one, which we shall call c, with

$$c_{\sigma(i)} = +1 \text{ for } i \in \overline{1,K} \text{ and } c_{\sigma(i)} = -1, \quad i \in \overline{K+1,N}. \tag{7}$$

where σ is the permutation defined by (1).

Before proceeding, let us restrict the parameters a, p,and d in (5) and (6) , and introduce a new parameter δ as follows:

R1 : $a > 0,\ p < 0,$ and $a < -p$.

R2 : δ satisfies $0 < \delta < -a - p$.

R3 : For all i, $|d_i| \leq -p - a - \delta$.

Also, we introduce some useful notation:

$$\epsilon_1 = \frac{\delta}{-a - p(N-1)},$$

$$\epsilon_2 = \frac{a}{a - p(N-1)}, \quad \epsilon_m = min(\epsilon_1, \epsilon_2).$$

These are all in the interval $(0,1)$ if R1, R2 and R3 hold. For $\epsilon < \epsilon_m$, let

$$\lambda_1(\epsilon) = \frac{\ell}{2(\delta + \epsilon[a - p(N-1)])} \ell n(2\epsilon^{-1} - 1),$$

$$\lambda_2(\epsilon) = \frac{\ell}{\epsilon(2-\epsilon)(a-p)},$$

$$\lambda_3(\epsilon) = \frac{\ell}{a} \ell n \frac{4[a - p(N-1)]}{ae}, \quad \lambda_4(\epsilon) = \frac{\ell}{2a} \ell n(2\epsilon^{-1} - 1).$$

We have defined ϵ_m, and now for $\epsilon \in (0, \epsilon_m)$ define λ_m by $\lambda_m = \max(\lambda_1(\epsilon).\lambda_2(\epsilon), \lambda_3, \lambda_4(\epsilon))$.

THEOREM 2 *Let R1, R2 and R3 hold. Suppose* $\epsilon \in (0, \epsilon_m)$, *and* $\lambda > \lambda_m$, *Suppose* $K \in \overline{1, N-1}$, *and the input list* d *is* σ*-ordered.*

There exists a $t^* > 0$ *such that for* $t > t^*$, *we have*

$$\begin{aligned} &v^\lambda_{\sigma(1)}(t) > \ldots > v^\lambda_{\sigma(K)}(t) > 1 - \epsilon > 0 \\ &> -1 + \epsilon > v^\lambda_{\sigma(K+1)}(t) > \ldots > v^\lambda_{\sigma(N)}(t) \end{aligned} \tag{8}$$

This theorem shows that at any time after t^*, the network has the KWTA property (13). The realisation of this circuit requires a mechanism to stop after all the output voltages have reached $1 - \epsilon$ or $-1 + \epsilon$.

The next point of this section concerns the fact that having processed a list d, the network must be reset to 0 before a second list can be processed. We change d to a vector with all components equal to $p(2K - N)$ so that $b_i = 0$ for all i, and 0 is an equilibrium point. But, unfortununately, we have found by simulation that for some $\lambda > \lambda_m$ the network may not return to zero when we switch off the inputs. Thus, we are led to find a new gain for which the network will reset and for which 0 is stable. In fact, the next result tells us something more than this, namely, that using a new gain, the zero state is the only equilibrium, and we will reach it from any other point.

PROPOSITION 1 *If R1 holds and* $\lambda < \frac{\ell}{a - p(N-1)} = \lambda_r$, *then* 0 *is a global asymptotically stable equilibrium point for*

$$\dot{u} = s[-\ell u + TG(\lambda u) + 0].$$

From the engineering point of view, changing the gain in order to reset is a drawback. It needs some additional circuits to implement and it will affect the processing time when repeated lists of variables are to be sorted.

4 Conclusions

This paper proposed a new continuous time Hopfield network which has the $KWTA$ property. The main emphasis is on proving that the model works as intended. It is shown that whenever the gain λ surpasses an easily computable value, the network moves into the set of those states in which all outputs v_i have values greater than $1-\epsilon$ or less than $-1+\epsilon$, and stays there. This matches well with the fact that large gain neurons are easily implemented in $CMOS$ technology. This network has the appealing property of infinite resolution, i.e. so long as the components of d are distinct, so that we can always conceive of the K largest of them, our network finds these K largest. Other networks have had a lower bound on the diffference between components of d, e.g. [11]. Let us also remark that our network does not require $N(N-1)$ connections between different neurons, although T is a full matrix. It has been shown in [8] that only $3N$ connections are needed, because of the regularity of the interconnection matrix, T. It is also worth mentioning that the computed gains λ_m and λ_r do not depend on K. This could be an advantage for applications where from succesive lists of N variables, we would want to find the largest K items, with K varying but N remaining the same.

References

[1] B. Calvert and C. Marinov, "Another K-winners-take-all neural network", IEEE Trans. Neural Networks, vol. 11, no.4, pp. 829-838, 2000.

[2] J. Choi and B.J. Sheu, "A high precision VLSI winner-take-all circuit for self organising neural networks", IEEE J. of Solid State Circuits, vol. 28, no. 5, pp. 576-594, 1993.

[3] K. Urahama and T. Nagao, "K-winners-take-all circuit with $O(N)$ complexity", IEEE Trans. Neural Networks, vol. 6, pp. 776-778, 1995.

[4] J.C. Yen, J.I. Guo and H-C. Chen, "A new k-winners-take all neural network and its array architecture", IEEE Trans. Neural Networks, vol, 9, no. 5, pp. 901–912, 1998.

[5] M. Lemmon and B.V.K. Vijayakumar, "Competitive learning with generalised winner-take-all activation", IEEE Trans. Neural Networks, vol. 3, pp. 167-175, 1992.

[6] S. Kaski and T. Kohonen, "Winner-take-all networks for physiological models of competitive learning", Neural Networks, vol. 7, pp. 973–984, 1994.

[7] T.X. Brown, "Neural networks for switching" in B. Yuhas and N. Ansari, editors Neural Networks in Telecommunications, Kluwer Academic Publ., Boston, London, Dordrecht, 1994.

[8] E. Majani, R. Erlanson and Y. Abu.Mostafa, "On the K-winners-take-all network", in D.S. Touretzky (ed), Advances in neural information processing systems, vol 1, Morgan Kaufmann, San Mateo, CA, pp. 634–642, 1989.

[9] R. Erlanson and Y. Abu-Mostafa, "Analog neural network as decoders", in D.S. Touretzky (ed.), Advances in neural information processing systems, vol 3, Morgan Kaufmann, Los Altos CA, pp. 585–588, 1991.

[10] J. Lazzaro, S.Ryckebusch, M.A. Mahowald and C.A. Mead, "Winner-take-all networks of $O(N)$ complexity", in D.S. Touretzky (ed), Advances in neural information processing systems, vol. 1, Morgan Kaufmann, San Mateo, CA, pp. 703–711, 1989.

[11] Y. Fang, M.A. Cohen and T.G. Kincaid, "Dynamics of a winner-take-all neural network", Neural Networks, vol. 9, pp. 1141–1154, 1996.