ELSEVIER

Information Processing Letters 54 (1995) 317-322

Information Processing Letters

On the performance guarantee of neural networks for NP-hard optimization problems

V. Zissimopoulos [1]

LRI, CNRS-URA 410, Bât 490, Université de Paris XI, 91405 Orsay Cedex, France

Communicated by L. Boasson; received 1 February 1995; revised 13 March 1995

Abstract

We give polynomial size threshold neural networks and encoding formalisms, which guarantee worst case performance for two hard optimization problems. We show that a massively parallel algorithm based on such neural network models guarantee an approximation ratio, asymptotically equal to $\Delta/2$ for the maximum independent set problem, where Δ is the maximum degree of the graph, and equal to 2 for the vertex covering problem. These results on the power of polynomial size threshold neural networks within polynomial number of neural updates provide the first approximation results for neural network models.

Keywords: Combinatorial problems; Analysis of algorithms; Combinatorial optimization; Neural networks; Maximum independent set; Heuristics; Approximation algorithms; Worst-case analysis

1. Introduction

Owing to their convergence properties, massive parallelism, generality, flexibility and adaptability, neural networks have drawn considerable attention in recent years from both academic and industrial communities. Several neural networks have been synthesized to solve approximately hard optimization problems. The empirical investigations indicate a very good performance of these models when they are compared to conventional heuristics. In general, in these studies, execution time is neglectable, expecting optical devices implementations allowing the models to operate at a higher speed than conventional electronics.

Recently, two theoretical results on the power of polynomial size threshold neural networks have been established. Bruck and Goodman [2] have shown that for any NP-hard optimization problem, a polynomial size threshold neural network that solves it, does not exist unless NP = co-NP. Also, finding ε-approximate solutions to the traveling salesman problem is not possible unless P = NP. Furthermore, Yao [5] has shown that for minimum set covering, maximum independent set, minimum vertex cover, maximum clique, maximum set packing and knapsack problems, getting ε-approximate solutions by a polynomial size neural network is impossible unless NP = co-NP.

In this paper, we show rigorously that neural networks have important approximation properties and thus, the empirical investigation is not the only way to study their performance. To this aim, we consider the maximum independent set problem (IS), i.e. finding in a simple graph $G(V,E)$ of order n and $|E| = |\{(u,v) \mid u,v \in V\}|$ edges, a subset V' of maximum cardinality such that $\forall u,v,\ u,v \in V' \Rightarrow (u,v) \notin E$.

[1] Email: vassilis@lri.lri.fr.

0020-0190/95/$09.50 © 1995 Elsevier Science B.V. All rights reserved
SSDI 0020-0190(95)00051-8

The best approximation ratio known for this problem is (asymptotically) equal to $\Delta/2$ [4], where Δ is the maximum degree of the graph. More precisely, for a connected graph the approximation ratio is shown to be equal to $(\Delta/2)(1+1/(n-1))+1/2$. We show that it is possible to design appropriate neural network topologies and to define encoding formalisms which permit to guarantee for the *IS* a level of performance and even more an approximation ratio equal, for large values of n and Δ, to the best actually known approximation ratio of sequential algorithms for *IS*. In fact, at a first time we present a simple neural network with an encoding formalism that guarantees approximation ratio equal to Δ. This topology was first presented by Aarts in [1]. Next, we give a new topology with an encoding formalism allowing to guarantee an approximation ratio equal asymptotically to $\Delta/2$. The neural network model that we propose is a massively parallel implementation of the algorithm presented in [4]. It is a discrete time system which can be described as a weighted undirected graph. The computational process carried out by the model is strongly related to a summarizing function, the so called networks consensus. Information can be stored via connection strengths between processing elements and then it can be retrieved by allowing the system to settle into local consensus maxima. For solving optimization problems this function can be associated with the objective function and the constraints of the problem. The evolution in time of the model promotes the optimization of this function.

Let n be the number of neurons or processing elements. The neural network is uniquely defined by a symmetric matrix $W=(w_{ij})_{n\times n}$, where w_{ij} is the connection strength between neurons i and j, and w_{ii} is an external or bias connection. The state of the network at time t is represented by a vector $U(t)=(u_i(t))_n$, where $u_i(t)\in\{0,1\}$ is the state of neuron i (OFF or ON). The consensus function is $C(t)=\sum_{ij} w_{ij}u_i(t)u_j(t)$, where i and j range over all pairs such that there exists a connection between neurons i and j. The next state of neuron i is computed as follows:

$$u_i(t+1)=\begin{cases}1-u_i(t) & \text{if } \Delta C_i(t)\geqslant 0,\\ u_i(t) & \text{otherwise,}\end{cases} \tag{1}$$

where ΔC_i is given by

$$\Delta C_i(t)=(1-2u_i(t))\Big[\sum_{j,j\neq i} w_{ij}u_j(t)+w_{ii}\Big]. \tag{2}$$

The next global state of the network $U(t+1)$ is computed from the current state by performing the evaluation (1) at a subset of vertices of the network, denoted by P. According to the size of the set P in each time interval the following three modes of operation are determined: serial mode when $|P|=1$, fully parallel mode when $|P|=n$ and limited parallel mode when $1<|P|<n$. A state vector $U(t)$ is called stable if there is no change in the state of the network. One of the most important properties of the model is the fact that it always converges to a stable state while operating in a serial mode. This stable state corresponds to a local maximum of the consensus function. This suggests the use of the network as a device for performing a local search for getting a maximal value of the objective function.

In this paper, we study neural networks operating in a serial mode ($|P|=1$) with threshold output function (Eq. 1). We give two neural network topologies and appropriate encoding formalisms and we show that a performance guarantee is achieved and even this equals the performance of the best sequential approximate algorithms, for *IS*.

2. Neural network topologies and performance guarantee

Let $G(V,E)$ be an undirected connected graph, where $V=\{v_1,v_2,\dots,v_n\}$ is the set of vertices and E is the set of edges, with $|V|=n$ and $|E|=m$. Edges are indicated by pairs of vertices, for example (v_i,v_j), the maximum degree by Δ, the minimum degree by δ and the degree of vertex v_i by Δ_i. Consider a neural network topology $T_1(U,C)$ isomorphic to the graph G, with $U=\{1,2,\dots,i,\dots,n\}$ the set of neurons, and C the set of connections. The state of a neuron i at time t is indicated by $u_i(t)$ or simply by u_i.

The encoding for the general case where weights are associated to the vertices of the graph is performed by the symmetric connection matrix $W=(w_{ij})$, $w_{ij}=c_i$ if $i=j$ and $w_{ij}=-[\max\{c_i,c_j\}+\varepsilon]\delta_{ij}$ if $i\neq j$, where $\delta_{ij}=1$ if $(v_i,v_j)\in E$ and 0 otherwise, and ε is a small real positive parameter. If there are not weights on the vertices we consider that $c_i=1$, $i=1,2,\dots,n$

and we get the same encoding matrix as proposed in [1]. The consensus of the network for each new configuration, is completely determined by the states of the neighbours of the neuron whose state is changed, say neuron i, and the corresponding strengths, according to (2) and the consensus of the current configuration. If the consensus of a given configuration of the network does not increase by any single state transition, i.e., $\Delta C_i \leqslant 0$ for all neurons, then the network is lying on a local maximum (stable state). The solution vector for *IS* is equal to the vector state of the network. In [1] it was proved that each stable state of this network with the above defined connection matrix (and with $c_i = 1, \forall i = 1, \ldots, n$) corresponds always to feasible and locally maximum solutions for the *IS*. This allows us to obtain the following result, for the maximum cardinality independent set problem.

Theorem 1. *Given a connected graph $G(V, E)$ with maximum degree Δ, the neural network $T_1(U, C)$ which is isomorphic to G, guarantees, for the maximum independent set problem on G, an approximation ratio equal to Δ, when it operates in a serial mode with threshold output function.*

Proof. In a stable state an active neuron prevents at most Δ neurons from being active. Note that, when a neuron becomes active, it remains always in this state, while its neighbours are definitely removed and they remain inactive. So, if $u_i = 1$ then $u_j = 0, \forall j \in N(i)$, where $N(i)$ is the set of neighbours of neuron i. The solution for the maximum *IS* is $\{v_i \mid u_i = 1 \wedge i \in \{1, \ldots, n\}\}$. In the worst case, $\lceil n/(\Delta+1)\rceil$ neurons are active providing a solution of size $h \geqslant n/(\Delta+1)$. Let us now consider an optimal solution including α vertices. The n vertices of the graph G can be partitioned into two sets S and $V-S$, with $|S| = \alpha$ and $V-S$ including the vertices discarded by the α vertices. Then, since all edges are either between S and $V-S$ or inside $V-S$, we have:

$$
\begin{aligned}
&\sum_{i\in S} \Delta_i \leqslant \sum_{j\in V-S} \Delta_j \\
&\quad\Rightarrow\quad |S|\delta \leqslant |V-S|\Delta \\
&\quad\Rightarrow\quad \alpha\delta \leqslant (n-\alpha)\Delta \\
&\quad\Rightarrow\quad \alpha \leqslant \Delta n/(\Delta+\delta).
\end{aligned}
$$

Finally, we have that $\alpha/h \leqslant \Delta(\Delta+1)/(\Delta+\delta) \leqslant \Delta$, i.e., the neural network T_1 guarantees an approximation ratio equal to Δ. $\square$

Let us consider the following topology: $T(U_1 \cup U_2 \cup U_2' \cup \{d\},\ C_1 \cup C_2 \cup C_2' \cup C_3 \cup C_3')$ including two levels of neurons $U_1, U_2 \cup U_2'$ and a single decision neuron $\{d\}$. The first level has $|U_1| = n$ neurons and $|C_1| = m$ inhibitory connections representing exactly the topology T_1 defined previously.

The second level includes two sub-networks with U_2, U_2' neurons and C_2, C_2' connections, respectively. Each neuron in U_2 corresponds to an edge of G and each inhibitory connection in C_2 links two neurons i and j if the corresponding edges, say $(v_k, v_l), (v_{k'}, v_{l'}) \in E$, $k \neq l$, $k' \neq l'$, have a common vertex. Let denote this part by T_2. Next, we refer to this part, as the first sub-network of the second level. T_2 is obviously, isomorphic to the line graph of G. Each neuron in U_2' corresponds to a vertex of G and each excitatory connection in C_2' links a neuron $i \in U_2'$ to a neuron $j \in U_2$ if and only if the vertex v_i is incident with the edge corresponding to neuron j. Thus, each neuron in U_2 is connected to exactly two neurons in U_2', while inside U_2' there is no connection at all. Next, the set of neurons U_2' with the C_2' connections to T_2 is denoted by (T_2') and is refered as the second sub-network of the second level.

The neuron d is a decision neuron without bias connection. It is linked by a set C_3 of inhibitory connections to all neurons in T_1, and with a set C_3' of excitatory connections to all neurons in T_2'. Between neurons of the two levels there is no connection at all.

On this topology, we define the following connection strengths,

$$
w_{ij} = \begin{cases}
\Delta_i^{-\Delta_i} & \text{if } i = j \wedge i \in U_1 \\
 & \quad \{\text{bias connections}\}, \\
-[\min\{\Delta_i^{-\Delta_i}, \Delta_j^{-\Delta_j}\} + \varepsilon]\delta_{ij} & \\
 & \text{if } i \neq j \wedge i, j \in U_1, \\
1 & \text{if } i = j \wedge i \in U_2 \\
 & \quad \{\text{bias connections}\}, \\
-[1+\varepsilon] & \text{if } i \neq j \wedge i, j \in U_2 \\
 & \quad \wedge\, v(i) \cap v(j) \neq \emptyset, \\
1 & \text{if } i \in U_2 \wedge j \in U_2', \\
1 - u_i & \text{if } i \in U_2' \wedge j = d, \\
-1 & \text{if } i \in U_1 \wedge j = d,
\end{cases} \tag{3}
$$

where $\delta_{ij} = 1$ if there is an edge $(v_i, v_j) \in E$ and 0 otherwise, ε is a real positive parameter such that $\varepsilon \leqslant [\Delta(\Delta+1)]^{-(\Delta+1)}$, $v(i)$ denotes the two vertices incident with the edge i and u_i is the state of the neuron i.

We consider now that the two levels T_1 and (T_2, T_2') operate in parallel. In the second level, the two sub-networks T_2 and T_2' operate sequentially. The sub-network T_2' operates once T_2 has reached stability. The connection matrix in each level being symmetric and since a serial mode of operation is assumed, the network reaches stability. Then, the output of the neurons in T_1 and T_2' pass to the decision neuron d which has two states 0 and 1. The state of this neuron is determined by its total input: $I_d = \sum_{j \in U_1 \cup U_2'} w_{dj} u_j$. If $I_d > 0$, the neuron becomes active and the solution for the *IS* is equal to the inactive neurons in T_2'. Otherwise, the neuron is inactive and the solution is obtained by the active neurons in T_1. Essentially, the state of the decision neuron determines the bigger solution for *IS*.

In the sequel, we consider the consensus function for each level, since the two levels operate in parallel and no connections exist between the two levels. The decision neuron is considered only after stability of both levels.

Lemma 2. *Each stable state of the neural network T, with the encoding matrix* (3), *corresponds to two feasible solutions, at least one of which is a local maximum for the independent set problem.*

Proof. The sub-network T_2 in the second level gives a locally maximum *IS* for the line graph of G. In fact, this topology and the encoding matrix are defined as in [1] and the proof can be found there. This independent set on the line graph of G provides a locally maximum matching for G. Next, after operation of the sub-network T_2', the active neurons in T_2' represent exactly the incident vertices with the matched edges (active neurons in T_2). Therefore, the inactive neurons in T_2' are the exposed vertices of G with respect to the considered matching (vertices that are not incident to any matched edge) which form a feasible solution for the *IS* in G [3]. In fact, if this solution is not a feasible solution, the matching would not be locally maximum.

Let us now revisit the first level. We prove that a feasible and locally maximum solution is obtained. Consider a configuration k of the network inducing a non feasible solution x^k. We prove that this configuration is not a stable state. In fact, since x^k is not feasible there are neurons i and j which are adjacent and both are on state ON. By changing the state of the neuron with the larger degree ($\Delta_i > \Delta_j$) the consensus increases. Really,

$$\begin{aligned}
\Delta C_i(k) &= C(k : u_i = 0) - C(k : u_i = 1) \\
&= -\Big(w_{ii} + \sum_{l=1,\, l\neq i}^{n} w_{il} u_l\Big) \\
&\geqslant -(w_{ii} + w_{ij}) \\
&\geqslant -(w_{ii} + [-\min\{w_{ii}, w_{jj}\} - \varepsilon]) \\
&= -(\Delta_i^{-\Delta_i} + [-\Delta_i^{-\Delta_i} - \varepsilon]) \\
&= \varepsilon > 0.
\end{aligned}$$

In the case where the two neurons have the same degree, also by changing the state of one of them the consensus increases. Moreover, if x^k does not correspond to a local maximum *IS*, then there is an inactive neuron i which is not adjacent with any active neuron. By changing the state of i the consensus increases, since $\Delta C_i = w_{ii} = \Delta_i^{-\Delta_i} > 0$. $\square$

Lemma 3. *For an irregular graph, at each stable state of the neural network T with encoding matrix* (3), *at least one neuron corresponding to a vertex with minimum degree is at state "N s" in the first level, and thus, at least one vertex with minimum degree is taken into account in the solution for the maximum independent set problem.*

Proof. Let us consider a stable state where all neurons corresponding to vertices with minimum degree $\delta < \Delta$ are OFF. Let i be one of these neurons. Consider the worst case where all neighbors of i are ON. The vertices corresponding to these neurons have degrees Δ_j, $j = 1, \ldots, \delta$ such that: $\delta < \Delta_1 \leqslant \Delta$, $\delta < \Delta_2 \leqslant \Delta$, $\ldots$, $\delta < \Delta_\delta \leqslant \Delta$. This state is not a stable state, since by changing the state of the neuron i the consensus increases. In fact,

$$\begin{aligned}
\Delta C_i(k) &= C(k : u_i = 1) - C(k : u_i = 0) \\
&= (+1)\Big(w_{ii} + \sum_{l=1,\, l\neq i}^{\delta} w_{il} u_l\Big)
\end{aligned}$$

$$
\begin{aligned}
&= \frac{1}{\delta^\delta} + \sum_{l=1}^{\delta}\Big[-\Big(\frac{1}{\Delta_l^{\Delta_l}} + \varepsilon\Big)\Big] \\
&\geqslant \frac{1}{\delta^\delta} - \Big[\sum_{l=1}^{\delta}\frac{1}{(\delta+1)^{\delta+1}} + \delta\varepsilon\Big] \\
&= \frac{1}{\delta^\delta} - \frac{\delta}{(\delta+1)^{\delta+1}} - \delta\varepsilon \\
&\geqslant \frac{\delta}{\delta^{\delta+1}} - \frac{\delta}{(\delta+1)^{\delta+1}} - \frac{\delta}{[\Delta(\Delta+1)]^{\Delta+1}} \\
&\geqslant \frac{\delta}{\delta^{\delta+1}} - \frac{\delta}{(\delta+1)^{\delta+1}} - \frac{\delta}{\delta^{\delta+1}(\delta+1)^{\delta+1}} \\
&= \frac{\delta^\delta C^1_{\delta+1} + \cdots + \delta C^\delta_{\delta+1}}{\delta^\delta(\delta+1)^{\delta+1}} > 0. \qquad \Box
\end{aligned}
$$

Note that, once a neuron corresponding to a vertex of minimum degree becomes active, it always remains active. Its neighbours are definitely eliminated even if they have the same degree. Lemma 2 and 3 show that the sub-network of the first level of neural network T presents a massive parallel implementation of the natural greedy algorithm of smallest degree [3].

Theorem 4. *Given a connected graph $G(V,E)$ with maximum degree Δ, the neural network T, with the encoding matrix* (3), *guarantees an approximation ratio asymptotically equal to $\Delta/2$ for the maximum independent set problem on G, when it operates in a serial mode with threshold output function.*

Proof. If the graph is regular, in the first level T_1 has bias connection w_{ii} equal to $\Delta^{-\Delta}$ for all neurons. The active neurons provide a solution h such that $h \geqslant n/(\Delta+1)$. So, by the expression for $\alpha \leqslant \Delta n/(\Delta+\delta)$, we get for regular graphs $\alpha \leqslant n/2$ and thus, $\alpha/h \leqslant (\Delta+1)/2$, i.e., an approximation ratio equal to $(\Delta+1)/2$.

We consider now irregular graphs ($\delta < \Delta$). The sub-network T_2 after stability provides a locally maximum matching m'. Since, $m' \leqslant n/2$, we can put $m' = (n/2)(1-\gamma)$, for a real $\gamma \in [0,1]$. Obviously, the $2m'$ vertices incident upon to the matched edges constitute a vertex covering C in G (a set of vertices such that $(u,v) \in E \Rightarrow u \in C$ or $v \in C$). These vertices correspond to the active neurons in T_2'. From this vertex covering we can obtain an independent set of size $n - |C|$ by a simple set difference. On the other hand, the optimal vertex covering is always greater or equal to $\frac{1}{2}|C|$. In particular, for $|C| = 2m'$ we obtain in T_2' an independent set (vertices corresponding to the inactive neurons) of size $h_2 = n-2m' = n-2((n/2)(1-\gamma)) = \gamma n$, with $\gamma \in [0,1]$. For the size of a global optimal maximum independent set we have: $\alpha \leqslant n - \frac{1}{2}|C| = n - m' = n - \frac{n}{2}(1-\gamma) = \frac{n}{2}(1+\gamma)$. Therefore, the second level of the neural network T gives: $\frac{\alpha}{h_2} \leqslant \frac{(1+\gamma)}{2\gamma}$.

In the first level, since the vertex with the smallest degree is included in the solution (lemma 2), after stability T_1 gives a solution of size

$$
\begin{aligned}
h_1 &\geqslant \frac{n-(\delta+1)}{\Delta+1} + 1 \\
&= \frac{n-(\delta+1)+\Delta+1}{\Delta+1} \geqslant \frac{n+1}{\Delta+1}.
\end{aligned}
$$

Clearly, by using the expression for α we obtain:

$$
\frac{\alpha}{h_1} \leqslant \frac{n/2(1+\gamma)}{(n+1)/(\Delta+1)} = \frac{n(1+\gamma)(\Delta+1)}{2(n+1)}.
$$

By the same way as in [4] the two functions of γ (upper bounds of α/h_2 and α/h_1), the first being monotonously decreasing and the second one monotonously increasing, have a "breakpoint" at $\gamma_0 = \frac{n+1}{n(\Delta+1)}$. For $\gamma \geqslant \gamma_0$ take the solution in T_2' while for $\gamma \leqslant \gamma_0$ take the solution in T_1. By replacing γ_0 in these functions we obtain an approximation ratio equal to $\frac{\Delta+1}{2}[1 - \frac{1}{n+1}] + \frac{1}{2}$. So, in the worst case, since the decision neuron chooses the best solution $h = \max\{h_1, h_2\}$, we have always the same approximation ratio which for large values of n and Δ tends to $\Delta/2$. $\Box$

In what concerns the architectural complexity of the neural network T, we remark that T_1 has n neurons, T_2 has m neurons, T_2' has n neurons and a decision neuron. In total, T has $2n + m + 1$ neurons. Also, T requires: m inhibitory connections and n bias connections in T_1, at most Δm inhibitory connections and m bias connections in T_2, $2m$ excitatory connections between T_2 and T_2', n excitatory connections between the decision neuron and T_2' and n inhibitory connections between the decision neuron and T_1. The total number of connections is at most $(\Delta+4)m + 3n$ which is bounded by $(n+4)m + 3n$.

If we consider a specified order to update neurons, the number of neural updates required by T to reach stability is at most equal to n^2. In fact, in T_1, a neuron corresponding to a vertex of minimum degree is activated after at most n neural updates (one cycle). More precisely, the first neuron, say i, corresponding to a vertex of minimum degree, becomes active the first time we meet it in the sequence and it always remains at this state. Also, the other neurons corresponding to vertices of minimum degree, if they are neighbours of already active neurons of minimum degree they become definitely inactive. Otherwise, they become definitely active. Thus, the neuron i reaches a definitive state after at most n neural updates. In the next cycle, at least the neighbours of the neuron i adjust their states and they become definitely inactive. In each new cycle at least one neuron, from the $n - (\delta + 1)$ remaining neurons, adjusts definitely its state. Finally, all neurons find their definitive state after at most $n - (\delta + 1) + 2 \leqslant n$ cycles. Therefore, T_1 reaches stability after at most n^2 neural updates. T_2 requires only one cycle to reach stability, i.e., m neural updates. This is because each neuron reaches its definitive state after only one neural update and if it becomes active, it eliminates definitely its neighbours. T_2' requires also only one cycle and thus, n neural updates. Consequently, since the two levels operate in parallel the neural updates required for taking a solution for the *IS* is equal to $\max\{n^2, m + n\}$.

Note that, in the proofs of Lemma 2 and Theorem 4 it is also shown that the sub-networks T_2 and T_2' in the second level of T represent a neural network for minimum vertex covering, which with the encoding matrix (3) guarantees an approximation ratio for this problem equal to 2. This neural network has $m + n$ neurons and at most $m(\Delta + 3) \leqslant m(n + 3)$ connections. The number of neural updates required for taking a solution for the vertex covering problem giving an approximation ratio equal to 2 is $m + n$. The developed algorithm on this neural network constitutes a massive parallel implementation of the greedy algorithm described in [3] which guarantees the same approximation ratio for minimum vertex covering problem.

References

[1] E. Aarts and J. Korst, *Simulated Annealing and Boltzmann Machines. A stochastic Approach to Combinational Optimization and Neural Computing* (John Wiley & Sons, New York, 1989).

[2] J. Bruck and J.W. Goodman, On the power of neural networks for solving hard problems, *J. Complexity* **6** (1990) 129–135.

[3] C.H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity* (Prentice Hall, Englewood Cliffs, NJ, 1981).

[4] V.Th. Paschos, A $(\Delta/2)$-approximation algorithm, for the maximum independent set problem, *Inform. Process. Lett.* **44** (1992) 11–13.

[5] X. Yao, Finding approximate solutions to NP-hard problems by neural networks is hard, *Inform. Process. Lett.* **41** (1992) 93–98.