Memorandum COSOR 92-14

The construction of minimal multi-layered perceptrons: a case study for sorting

P.J. Zwietering
E.H.L. Aarts
J. Wessels

Eindhoven, June 1992
The Netherlands
The construction of minimal multi-layered perceptrons: a case study for sorting

P.J. Zwietering\textsuperscript{1}, E.H.L. Aarts\textsuperscript{1,2} and J. Wessels\textsuperscript{1,3}

1. Eindhoven University of Technology, Department of Mathematics and Computing Science, P.O. Box 513, 5600 MB Eindhoven, the Netherlands
2. Philips Research Laboratories, P.O. Box 80.000. 5600 JA Eindhoven, the Netherlands
3. International Institute for Applied Systems Analysis, A-2361 Laxenburg, Austria

Abstract

We consider the construction of minimal multi-layered perceptrons for solving combinatorial optimization problems. Though general in nature, the construction method is presented as a case study for the sorting problem. The presentation starts with an $O((n!)^2)$ three-layered perceptron based on complete enumeration, that solves the sorting problem of $n$ numbers. This network is then gradually reduced to an $O(n^2)$ three-layered perceptron, which can be viewed as a neural implementation of Preparata’s parallel enumerative sorting algorithm.

Key words: minimal multi-layered perceptrons, feed-forward neural networks, combinatorial optimization problems, sorting
1 Introduction

Finding a feed-forward neural network that solves a given problem is often based on the following trial and error approach. A certain network configuration is chosen and next, the error back-propagation algorithm proposed by Rumelhart, Hinton & Williams [6] is used, until a satisfactory result is obtained. In order to arrive at a more methodical approach for the network construction, we consider the task of finding a multi-layered perceptron of minimal size that solves a given problem. The class of combinatorial optimization problems is chosen to serve as a test bed for developing such a methodical approach. The reason for this is that many problems found in OR-practice can be formalized as a combinatorial optimization problem. In a previous paper we showed that a large subclass of the class of combinatorial optimization problems can be solved by a three-layered perceptron. Furthermore, we presented a general construction method for obtaining this three-layered perceptron; see Zwietering, Aarts & Wessels [7]. However, this method did not guarantee any minimality; neither with respect to the number of hidden layers, nor with respect to the number of used units. We therefore tried to find lower bounds on the number of layers and the number of units in each layer. For two specific problems we have been able to show that the minimal number of layers needed to solve the problem is indeed three: the sorting problem; see Zwietering, Aarts & Wessels [10], and the so-called Wagner & Whitin variant of the lot-sizing problem; see Zwietering, Aarts & Wessels [9]. Next, we tried to find a way to reduce the number of units used in a layer by layer approach. The first problem that could be handled in this way is the sorting problem and the results are presented in this paper. Recently, we have shown that these results can be generalized to reduce the size of the first layer for a large number of combinatorial optimization problems, and that the resulting number of first-layer units is provably minimal; see Zwietering, Aarts & Wessels [11].

The paper is organized as follows. In Section 2, we reformulate the problem of sorting \( n \) numbers as a combinatorial optimization problem and derive a three-layered perceptron with \( O((n!)^2) \) units, using the method presented in [7]. In Section 3 the size of the network is reduced to a network of size \( O(n^2) \) using a layer by layer approach. In Section 3.1 we reduce the size of the first hidden layer from \( \frac{1}{2}n!(n! - 1) \) to \( \frac{1}{2}n(n - 1) \). In Section 3.2 we show that the size of the second hidden layer can be reduced from \( n! \) to \( n^2 \). In Section 4 we discuss the relation between the obtained three-layered perceptron and the well-known parallel enumerative sorting algorithm proposed by Preparata. Section 5 concludes the paper with some final remarks.
2 Reformulation and first results

In this paper we consider the problem of sorting \( n \) numbers for some fixed \( n \in \mathbb{N} \). For any given array of numbers \( x_1, \ldots, x_n \in \mathbb{R} \), the objective is to find an ascending ordering of these numbers, i.e. we must find a permutation \( \pi \) of \( \{1, \ldots, n\} \) such that \( x_{\pi(i)} \leq x_{\pi(i+1)} \) for all \( i = 1, \ldots, n-1 \). In this way \( \pi(i) \) denotes the index of the number that must come at position \( i \) and, alternatively, \( \pi^{-1}(j) \) denotes the position of the \( j \)th number in the ordering of \( x_1, \ldots, x_n \). It is intuitively clear that the sorting problem is equivalent to the problem of finding a permutation \( \pi \) that \( \sum_{i=1}^{n} i x_{\pi(i)} \). In fact it is trivial to see that any \( \pi \) that maximizes \( \sum_{i=1}^{n} i x_{\pi(i)} \), induces an ascending ordering on the array of numbers \( x_1, \ldots, x_n \). That the reverse is also true is shown in Section 3.1.

From the above it follows that the problem of sorting \( n \) numbers can be formalized as a combinatorial optimization problem. Before doing so, we discuss an extension of the standard sorting problem which gives a better insight into our approach and leads to a useful formalization. Let \( \{\nu_i\}_{i=1}^{n} \) be a sequence of \( n \) numbers with \( \nu_i \neq \nu_j \) for all \( i \neq j \) (for brevity we write \( \{\nu_i\}_{i=1}^{n} \) instead of \( \{\nu_i\}_{i=1}^{n} \)). Then the problem of sorting \( x_1, \ldots, x_n \) according to \( \{\nu_i\} \) is to find a permutation \( \pi \) such that \( x_{\pi(i)} \leq x_{\pi(j)} \) for all \( i, j = 1, \ldots, n \) with \( \nu_i < \nu_j \). This problem is not harder than the standard problem: if \( \tilde{\pi} \) sorts \( x_1, \ldots, x_n \) standardly and \( \tilde{\pi} \) sorts \( \{\nu_i\} \), then \( \pi = \tilde{\pi} \tilde{\pi}^{-1} \) sorts \( x_1, \ldots, x_n \) according to \( \{\nu_i\} \).

In Section 3.1 we show that solving the extended sorting problem is equivalent to maximizing \( \sum_{i=1}^{n} \nu_i x_{\pi(i)} \), if \( \{\nu_i\} \) is a monotone sequence. Using the ‘two-phase’ sorting procedure with \( \tilde{\pi} \) and \( \tilde{\pi} \) similar as above, one can easily see that this implies that the extended sorting problem is equivalent to maximizing \( \sum_{i=1}^{n} \nu_i x_{\pi(i)} \) for all sequences \( \{\nu_i\} \) with \( \nu_i \neq \nu_j \) for \( i \neq j \).

Next, the extended sorting problem is formalized as a combinatorial optimization problem \((I, S, F, c)\), where \( I \) denotes the set of problem instances, \( S \) denotes the set of solutions, \( F \subseteq S \) denotes the set of feasible solutions, \( c \) denotes the cost function and where for a given instance \( x \in I \) the problem is to find a feasible solution \( y \in F \) with minimal cost \( c(y; x) \):

\[
I = \mathbb{R}^n, \\
S = \{\pi \mid \pi \text{ is an integer function on } \{1, \ldots, n\}\}, \\
F = \{\pi \mid \pi \text{ is a permutation of } \{1, \ldots, n\}\}, \\
c(\pi; x) = -\sum_{i=1}^{n} \nu_i x_{\pi(i)}. \tag{1}
\]

In the remainder of this section we describe the three-layered perceptron that is obtained by applying the general method given in [7] to the above combinatorial optimization problem, since it serves as a starting point for the three-layered perceptron with \( \mathcal{O}(n^2) \)
units that we derive in Section 3.

The first step is to put the combinatorial optimization problem \((I, S, F, c)\) in a 0–1-formulation. This is done by representing permutations of \(\{1, \ldots, n\}\) as \(n \times n\) 0–1-matrices: \(y \in \{0, 1\}^{n \times n}\) represents the permutation \(\pi\) if \(y_{ij} = 1_{\pi(i)=j}\) for all \(i, j\), where \(1_{\{\}\}\) denotes the true-false indicator \((1_{\text{true}} = 1, 1_{\text{false}} = 0)\). One can easily verify that for any sequence \(\{\alpha_i\}\) we have that \(\sum_{j=1}^{n} \alpha_j y_{ij} = \alpha_{\pi(i)}\), if \(y\) represents \(\pi\). Hence, we obtain the following 0–1-formulation of the extended sorting problem.

\[
I = \mathbb{R}^n, \\
S = \{0, 1\}^{n \times n}, \\
F = \{y \in \{0, 1\}^{n \times n} | \sum_{i=1}^{n} y_{ij} = \sum_{j=1}^{n} y_{ij} = 1\}, \\
c(y; x) = -\sum_{i=1}^{n} \sum_{j=1}^{n} \nu_i x_j y_{ij}, 
\]

where for a given \(x \in I\) the problem is to find \(y \in F\) that minimizes \(c(y; x)\).

The second step is to add a preference measure \(p : S \to \mathbb{R}\) to the problem: if for some instance there exists more than one feasible solution with minimal costs, we prefer the solution \(y \in F\) with the highest value of the preference measure \(p(y)\). For obvious reasons we assume that \(p\) satisfies \(p(y) \neq p(z)\) for all \(y \neq z\). Moreover, we assume here that \(p\) is of the form:

\[
p(y) = \sum_{i=1}^{n} \sum_{j=1}^{n} \xi_i \eta_j y_{ij}, 
\]

where \(\{\xi_i\}\) and \(\{\eta_i\}\) are two sequences with \(\xi_i \neq \xi_j\) and \(\eta_i \neq \eta_j\) for all \(i \neq j\). Note that this property is not sufficient for having \(p(y) \neq p(z)\) for all \(y \neq z\), and we therefore assume that \(\{\xi_i\}\) and \(\{\eta_i\}\) are monotone.

The third step is to verify that the combinatorial optimization problem \((I, S, F, (c, p))\) with additional preference measure \(p\), is equivalent to the classification problem \((\Omega, \Gamma)\), with \(\Omega = I, \Gamma = \{\Omega_y \mid y \in F\}\) and \(\Omega_y\) given by:

\[
\Omega_y = \{x \in I \mid \forall z \in F \setminus \{y\} : c(y; x) < c(z; x) \lor (c(y; x) = c(z; x) \land p(y) > p(z))\}, 
\]

where for each instance \(x \in \Omega\) the problem is to classify \(x\) as a member of one of the subsets \(\Omega_y\) for some \(y \in F\) (see also [7]).

In the final step we derive a three-layered perceptron that solves \((\Omega, \Gamma)\), and hence, solves \((I, S, F, (c, p))\). For this purpose we let \(\Theta\) denote the hard-limiting response function,

\footnote{To make (1) and (2) exactly equivalent, we have to impose some conditions on this function. However, since these conditions are irrelevant to the problem definition, they have been left out.}
which satisfies $\Theta(\lambda) = 1$ if $\lambda \geq 0$ and $\Theta(\lambda) = 0$ if $\lambda < 0$. For notational convenience we extend this definition to all vectors $\lambda$ of arbitrary dimension, with the assumption that $\Theta$ is applied element-wise.

**Theorem 1** Let $f : \mathbb{R}^n \rightarrow \{0, 1\}^{n \times n}$ be the function $f = g^{(3)} \circ g^{(2)} \circ g^{(1)}$, with $g^{(l)} = \Theta(\tilde{g}^{(l)})$ and $\tilde{g}^{(l)}$ for $l = 1, 2, 3$ defined by:

$$
\forall y, y' \in F, p(y) > p(y') : \tilde{g}^{(1)}_{y, y'}(x) = \sum_{k=1}^{n} \sum_{l=1}^{n} \nu_k(y_{kl} - y'_{kl})x_l,
$$

$$
\forall y \in F : \tilde{g}^{(2)}_{y}(w) = \sum_{z, z' \in F, p(z) > p(z')} (1_{z = y} - 1_{z' = y})w_{z, z'} - \sum_{z \in F, p(z) < p(y)} 1,
$$

(5)

$$
\forall i, j = 1, \ldots, n : \tilde{g}^{(3)}_{i, j}(v) = \sum_{z \in F} z_{ij}v_z - 1.
$$

Then $f$ represents a three-layered perceptron that solves $(\Omega, \Gamma)$.

**Proof** (Sketch, for details see [7])

Since $g^{(1)}$, $g^{(2)}$ and $g^{(3)}$ represent one-layered perceptrons and have matching dimensions, $f = g^{(3)} \circ g^{(2)} \circ g^{(1)}$ represents a three-layered perceptron.

Define $\Psi_{y, z} = \{x | c(y; x) \leq c(z; x)\}$, then one easily verifies that $x \in \Psi_{y, z}$ iff $g^{(1)}_{y, z}(x) = 1$. Furthermore, we have

$$
\Omega_y = \bigcap_{z \in F, p(z) < p(y)} \Psi_{y, z} \cap \bigcap_{z \in F, p(z) > p(y)} \Psi^*_{z, y},
$$

which implies $x \in \Omega_y$ iff $(\tilde{g}^{(2)} \circ g^{(1)})(x) = 1$. Due to the addition of the preference measure, the subsets $\Omega_y$ are disjoint. This can be shown to imply that $x \in \Omega_y$ iff $x \in V_{h_i}^{(ij)}$ for all $i, j = 1, \ldots, n$, where $V_{q}^{(ij)} = \bigcup_{z \in F, z_{ij} = q} \Omega_z$ and $(V_{q}^{(ij)})^* = V_{1-q}^{(ij)}$ for $q = 0, 1$ (see [7]). Hence, $x \in \Omega_y$ iff $(\tilde{g}^{(3)} \circ (g^{(2)} \circ g^{(1)}))(x) = y_{ij}$ for all $i, j = 1, \ldots, n$, which completes the proof of the theorem.

Since $|F| = n!$, the above three-layered perceptron has $\frac{1}{2}n!(n! - 1)$ units in the first hidden layer and $n!$ units in the second hidden layer. In the next section we show that this three-layered perceptron can be reduced to a three-layered perceptron with $\frac{1}{2}n(n - 1)$ units in the first hidden layer and $n^2$ units in the second hidden layer, giving a total of $\frac{1}{2}n(5n - 1)$ units (including the $n^2$ units in the output layer).

3 Main Results

Consider again the combinatorial optimization problem $(I, S, F, (c, p))$ given by (2) in the previous section with the additional preference measure $p$ given by (3). In this section
we rigorously derive a three-layered perceptron with $\frac{1}{2}n(5n - 1)$ units that solves the above problem, by reducing the size of the three-layered perceptron (5), presented in the previous section. This is done in two steps. Firstly, we show that the number of units in the first hidden layer can be reduced from $\frac{1}{2}n!(n! - 1)$ to $\frac{1}{2}n(n - 1)$; see Section 3.1. Secondly, we show that the number of units in the second hidden layer can be reduced from $n!$ to $n^2$; see Section 3.2.

In the derivations presented in this section, we will not use the 0-1-representation of a permutation, since it complicates the notation. Instead we return to the original formulation using $\pi$'s, i.e. we consider the combinatorial optimization problem $(I, S, F, (c, p))$ given by (1), where for a given $x \in I$ the problem is to find a $\pi \in F$ that minimizes $c(y; \pi)$. If there is more than one optimal solution, we choose the one with the highest value of the preference measure $p$ given by (cf. (3)):

$$p(\pi) = \sum_{i=1}^{n} \xi_i \eta_{x(i)}. \quad (6)$$

The corresponding classification problem $(\Omega, \Gamma)$ has classifying subsets given by (cf. (4)):

$$\Omega_\pi = \{ x | \forall \tilde{\pi} \in F \setminus \{ \pi \} : c(\pi; x) < c(\tilde{\pi}; x) \lor (c(\pi; x) = c(\tilde{\pi}; x) \land p(\pi) > p(\tilde{\pi})) \}. \quad (7)$$

In the following section we use the above formulation to find a multi-layered perceptron with $\frac{1}{2}n(n - 1)$ units in the first hidden layer, by deriving an alternative definition of the subsets $\Omega_\pi$.

### 3.1 Reducing The First Hidden Layer

The classifying subsets $\Omega_\pi$ given by (7) are defined using a total of $\frac{1}{2}|F|(|F| - 1) = \frac{1}{2}n!(n! - 1)$ bounding hyperplanes, which account for the $\frac{1}{2}n!(n! - 1)$ units in the first hidden layer (see the proof of Theorem 1). We need these many hyperplanes (units) since in general the costs of a permutation has to be compared with the costs of every other permutation and every comparison defines a hyperplane. In the first main result of this paper, we show that —for this cost function— the number of comparisons can be substantially reduced. We prove that we only have to compare the costs of a permutation to the costs of the permutations that are obtained by swapping two successive entries of that permutation, which yields a total of $\frac{1}{2}n(n - 1)$ comparisons. In order to formalize the above statement we give the following definitions.

**Definition 1** For all $k = 1, \ldots, n - 1$ the permutation $\pi^{(k)}$ is obtained from the permutation $\pi$ by swapping the $k^{th}$ and $(k + 1)^{th}$ entry:

$$\pi^{(k)}(i) = \begin{cases} 
\pi(i), & i \neq k, k + 1 \\
\pi(2k + 1 - i), & i = k, k + 1. 
\end{cases} \quad (8)$$
Definition 2 For every permutation \( \pi \) the set \( \Omega'_{\pi} \) is defined by:
\[
\Omega'_{\pi} = \{ x | \forall k : c(\pi; x) < c(\pi^{(k)}; x) \lor (c(\pi; x) = c(\pi^{(k)}; x) \land p(\pi) > p(\pi^{(k)})) \}. \tag{9}
\]

In Theorem 2 we prove that \( \Omega_{\pi} = \Omega'_{\pi} \) for all \( \pi \in F \), which is then used to show that all the subsets \( \Omega_{\pi} \) can be defined using a total of \( \frac{1}{2}n(n-1) \) hyperplanes only. The main idea in the proof of Theorem 2 is that we write the difference in costs of two permutations as a linear combination (Lemma 1), with positive multiplication factors (Lemma 2), of the difference in costs of the first permutation and a number (at most \( n - 1 \)) of successive swaps of this permutation necessary to obtain the second permutation. First, we need two more preliminary definitions.

Definition 3 For every pair of sequences \( \{\alpha_i\} \) and \( \{\beta_i\} \) the mapping \( s_{\alpha,\beta} : F \to \mathbb{R} \) is defined by:
\[
 s_{\alpha,\beta}(\pi) = \sum_{i=1}^{n} \alpha_i \beta_{\pi(i)}.
\tag{10}
\]
Notice that both \( c(\pi; x) \) and \( p(\pi) \) can be put in the form given by (10).

Definition 4 For every sequence \( \{\alpha_i\} \) with \( \alpha_i \neq \alpha_{i+1} \) and for all \( k = 1, \ldots, n-1 \) we define the mapping \( D^{(k)}_{\alpha} : F \to \mathbb{R} \) by:
\[
 D^{(k)}_{\alpha}(\pi) = (\alpha_k - \alpha_{k+1})^{-1} \sum_{i=1}^{k} (\alpha_i - \alpha_{\pi(i)}).
\tag{11}
\]

Lemma 1 Let \( \{\alpha_i\} \) and \( \{\beta_i\} \) be a pair of sequences with \( \alpha_i \neq \alpha_{i+1} \). Let \( \pi, \hat{\pi} \in F \), then we have:
\[
 s_{\alpha,\beta}(\pi) - s_{\alpha,\beta}(\pi \circ \hat{\pi}^{-1}) = \sum_{k=1}^{n-1} D^{(k)}_{\alpha}(\hat{\pi})(s_{\alpha,\beta}(\pi) - s_{\alpha,\beta}(\pi^{(k)})). \tag{12}
\]

Proof
Using Definitions 3, 4 and 1, the result follows using some elementary calculations:
\[
s_{\alpha,\beta}(\pi) - s_{\alpha,\beta}(\pi \circ \hat{\pi}^{-1}) =
\]
\[
= \sum_{i=1}^{n} \alpha_i \beta_{\pi(i)} - \sum_{i=1}^{n} \alpha_i \beta_{\pi(z-1(i))}
\]
\[
= \sum_{i=1}^{n} \alpha_i \beta_{\pi(i)} - \sum_{i=1}^{n} \alpha_{z(i)} \beta_{\pi(i)}
\]
\[
= \sum_{i=1}^{n} (\alpha_i - \alpha_{z(i)} \left[ \sum_{k=i}^{n-1} (\beta_{\pi(k)} - \beta_{\pi(k+1)}) + \beta_{\pi(n)} \right])
\]
\[
= \sum_{i=1}^{n} \sum_{k=i}^{n-1} (\alpha_i - \alpha_{z(i)})(\beta_{\pi(k)} - \beta_{\pi(k+1)}) + \sum_{i=1}^{n} (\alpha_i - \alpha_{z(i)}) \beta_{\pi(n)}
\]
Lemma 2 Let \( \{\alpha_i\} \) be a monotone sequence, then we have:

(i) \( D^{(k)}(\pi) \geq 0 \) for all \( \pi \in F \) and \( k = 1, \ldots, n-1 \).

(ii) \( D^{(k)}(\pi) = 0 \) implies \( D^{(l)}(\pi) = 0 \) for every sequence \( \{\alpha'_i\} \) with \( \alpha'_i \neq \alpha'_{i+1} \).

Proof

(i) Take \( \pi \in F, k \in \{1, \ldots, n-1\} \) and define \( A_k = \{\pi(1), \ldots, \pi(k)\}, B_k = \{1, \ldots, k\} \setminus A_k \) and \( C_k = A_k \setminus \{1, \ldots, k\} \). Then obviously \( |B_k| = |C_k| \) and \( (\alpha_k - \alpha_{k+1})^{-1}(\alpha_i - \alpha_j) > 0 \) for all \( i \in B_k \) and \( j \in C_k \). This implies:

\[
D^{(k)}(\pi) = (\alpha_k - \alpha_{k+1})^{-1} \left[ \sum_{i \in \{1, \ldots, k\}} \alpha_i - \sum_{i \in A_k} \alpha_i \right]
\]

\[
= (\alpha_k - \alpha_{k+1})^{-1} \left[ \sum_{i \in B_k} \alpha_i - \sum_{i \in C_k} \alpha_i \right]
\]

\[
\geq 0,
\]

with the equal sign only if \( B_k = C_k = \emptyset \). This completes the proof of (i).

(ii) From the proof of (i) above it follows that \( D^{(j)}(\pi) = 0 \) implies that \( B_k = \emptyset \) and hence, \( A_k = \{1, \ldots, k\} \). This yields that for every sequence \( \{\alpha'_i\} \) with \( \alpha'_i \neq \alpha'_{i+1} \) we have:

\[
D^{(k)}(\pi) = (\alpha'_k - \alpha'_{k+1})^{-1} \left[ \sum_{i \in \{1, \ldots, k\}} \alpha'_i - \sum_{i \in A_k} \alpha'_i \right] = 0,
\]

which completes the proof of the lemma.

\[\square\]

Theorem 2 Let \( \pi \in F, \Omega_\pi \) and \( \Omega'_\pi \) given by (7) and (9), respectively. Then \( \Omega_\pi = \Omega'_\pi \).

Proof

(\( \subseteq \)) Follows directly from the definitions (7) and (9).
(2) Let \( x \in \Omega_n \) and \( \hat{\pi} \in F \setminus \{\pi\} \). Define \( \hat{\pi} = \hat{\pi}^{-1} \circ \pi \), then we have \( \hat{\pi} = \pi \circ \hat{\pi}^{-1} \). By noting \( c(\pi; x) = -s_{\nu,x}(\pi) \), \( c(\hat{\pi}; x) = -s_{\nu,x}(\pi \circ \hat{\pi}^{-1}) \) and using Lemma 1 we find:

\[
c(\pi; x) - c(\hat{\pi}; x) = \sum_{k=1}^{n-1} D^{(k)}(\hat{\pi})(c(\pi; x) - c(\pi^{(k)}; x)).
\] (13)

From (13) combined with Lemma 2, part (i), we conclude two things. Firstly, since \( x \in \Omega_n \) implies \( c(\pi; x) \leq c(\pi^{(k)}; x) \) for all \( k \), we have \( c(\pi; x) \leq c(\hat{\pi}; x) \). Secondly, we find that \( c(\pi; x) = c(\hat{\pi}; x) \) implies \( c(\pi; x) = c(\pi^{(k)}; x) \) or \( D^{(k)}(\hat{\pi}) = 0 \) for \( k = 1, \ldots, n - 1 \). We complete the proof by showing that this implies \( p(\pi) > p(\hat{\pi}) \).

By noting \( p(\pi) = s_{\xi,\nu}(\pi), p(\hat{\pi}) = s_{\xi,\nu}(\pi \circ \hat{\pi}^{-1}) \) and using Lemma 1 we find:

\[
p(\pi) - p(\hat{\pi}) = \sum_{k=1}^{n-1} D^{(k)}(\hat{\pi})(p(\pi) - p(\pi^{(k)})).
\] (14)

Since \( x \in \Omega_n \) we have that \( c(\pi; x) = c(\pi^{(k)}; x) \) implies \( p(\pi) > p(\pi^{(k)}) \). Furthermore, from Lemma 2, part (ii), we know that \( D^{(k)}(\hat{\pi}) = 0 \) implies \( D^{(k)}(\hat{\pi}) = 0 \). Using these results in (14) we conclude that \( c(\pi; x) = c(\hat{\pi}; x) \) implies \( p(\pi) \geq p(\hat{\pi}) \). However, \( p(\pi) \neq p(\hat{\pi}) \) because \( \hat{\pi} \neq \pi \).

**Corollary 1** If the sequences \( \{\nu_i\}, \{\xi_i\} \) and \( \{\eta_i\} \), which define \( c \) and \( p \), respectively, are all increasing, then for all \( \pi \in F \) we have:

\[
\Omega_\pi = \{ x \mid \forall k : x_{\pi(k)} < x_{\pi(k+1)} \land (x_{\pi(k)} = x_{\pi(k+1)} \land \pi(k) < \pi(k+1)) \}. \] (15)

**Proof**

From Theorem 2 it follows that \( \Omega_\pi = \Omega'_\pi \), with \( \Omega'_\pi \) given by (9). Secondly, one easily verifies that \( c(\pi; x) < c(\pi^{(k)}; x) \) iff \( x_{\pi(k)} < x_{\pi(k+1)} \), if \( \{\nu_i\} \) is increasing, and \( p(\pi) > p(\pi^{(k)}) \) iff \( \pi(k) < \pi(k+1) \), if \( \{\xi_i\} \) and \( \{\eta_i\} \) are increasing.

This result also holds if \( \{\nu_i\} \) is increasing and \( \{\xi_i\} \) and \( \{\eta_i\} \) are both decreasing, as follows directly from \( p(\pi) = \sum_{i=1}^{n} \xi_i \eta_{\pi(i)} = \sum_{i=1}^{n} (-\xi_i)(-\eta_{\pi(i)}) \). Similarly, we have that if \( \{\nu_i\} \) is decreasing, the first smaller-than symbol in (15) has to be replaced by a greater-than symbol. Finally, if either \( \{\xi_i\} \) or \( \{\eta_i\} \) (not both) is decreasing, the second smaller-than symbol has to be replaced by a greater-than symbol.

In the remainder of this paper we assume that the three sequences are increasing, as the other cases are treated similarly. Using Theorem 2 and (15), it follows that the combinatorial optimization problem \( (I, S, F, (c, p)) \) given by (1) is equivalent to the problem of finding an ordered sorting of \( n \) given numbers, where —due to our assumption on \( \{\nu_i\} \), \( \{\xi_i\} \) and \( \{\eta_i\} \)— the objective is to find an ascending sorting of the given set of numbers in which the original order of the numbers is preserved as good as possible. In Theorem 3 we show how Corollary 1 can be used to obtain a second three-layered perceptron for the problem \( (I, S, F, (c, p)) \).
Theorem 3  Let $f : \mathbb{R}^n \rightarrow \{0, 1\}^{n \times n}$ be the function $f = g^{(3)} \circ g^{(2)} \circ g^{(1)}$, with $g^{(l)} = \Theta(g^{(l)})$ and $\tilde{g}^{(l)}$ for $l = 3, 2, 1$ defined by, respectively:

$$
\forall i < j : \tilde{g}^{(1)}_{ij}(x) = \sum_{k=1}^{n} (1_{k=j} - 1_{k=i})x_k,
$$

$$
\forall y \in F : \tilde{g}^{(2)}_y(w) = \sum_{k=1}^{n} \sum_{l=k+1}^{n} \sum_{m=1}^{n-1} (y_{mk}y_{m+1,l} - y_{ml}y_{m+1,k})w_{kl} - \sum_{k=1}^{n} \sum_{l=k+1}^{n} \sum_{m=1}^{n-1} y_{mk}y_{m+1,l},
$$

(16)

$$
\forall i, j : \tilde{g}^{(3)}_{ij}(v) = \sum_{z \in F} v_{iz} - 1.
$$

Then $f$ represents a three-layered perceptron that solves $(\Omega, \Gamma)$.

Proof (Analogous to the proof of Theorem 1)

Define $\Upsilon_{ij} = \{x \mid x_i \leq x_j\}$, then $x \in \Upsilon_{ij}$ iff $\tilde{g}^{(1)}_{ij}(x) = 1$ and from (15):

$$
\Omega_x = \bigcap_{k: \pi(k) < \pi(k+1)} \Upsilon_{\pi(k), \pi(k+1)} \cap \bigcap_{k: \pi(k) > \pi(k+1)} \Upsilon_{\pi(k+1), \pi(k)}.
$$

This implies $x \in \Omega_x$ iff $(\tilde{g}^{(2)}_y \circ g^{(1)})(x) \geq 0$, where

$$
\tilde{g}^{(2)}_y(w) = \sum_{k: \pi(k) < \pi(k+1)} (w_{\pi(k), \pi(k+1)} - 1) - \sum_{k: \pi(k) > \pi(k+1)} w_{\pi(k+1), \pi(k)}.
$$

It is straightforward to show that $\tilde{g}^{(2)}_y(w) = \tilde{g}^{(2)}_y(w)$ if $y$ and $\pi$ represent the same permutation, and hence, $x \in \Omega_y$ iff $(\tilde{g}^{(2)}_y \circ g^{(1)})(x) = 1$. The rest of the proof is equal to the last part of the proof of Theorem 1. \qed

The above theorem shows that the number of units in the first hidden layer of the three-layered perceptron that solves the problem, can be reduced from the $\frac{1}{2}n!(n! - 1)$ used in (5) to $\frac{1}{2}n(n-1)$ in (16). However, the size of the second hidden layer is still $|F| = n!$. The next step is therefore to reduce the size of the second hidden layer, which is the subject of the following section.

3.2 Reducing The Second Hidden Layer

The above results for the first hidden layer do not directly imply a method for reducing the number of units in the second hidden layer. We therefore need a new idea for reducing the second hidden layer. The key result is to derive an alternative expression for the sets $V_1^{(ij)} = \bigcup_{y \in F, y_{ij} = 1} \Omega_y = \bigcup_{x \in F, \pi(i-1) = j} \Omega_x$. using the expression (15) for $\Omega_x$. From this
alternative expression we then obtain the perceptron configuration with $O(n^2)$ units that solves $(I, S, F, (c, p))$; see Theorem 4. First we need a last definition.

**Definition 5** For all $x \in \mathbb{R}^n$ and $j = 1, \ldots, n$ we define the number $\Lambda_j(x) \in \{1, \ldots, n\}$ by:

$$\Lambda_j(x) = |\{k \mid x_k < x_j \lor (x_k = x_j \land k \leq j)\}|. \quad (17)$$

**Lemma 3** Let $i, j \in \{1, \ldots, n\}$ and $V_{i}^{(j)} = \bigcup_{\pi \in F, \pi(i) = j} \Omega_\pi$, where $\Omega_\pi$ is given by (15) and $\Lambda_j(x)$ is given by (17). Then:

$$V_{i}^{(j)} = \{x \mid \Lambda_j(x) = i\}. \quad (18)$$

**Proof**

($\subseteq$) Let $x \in \Omega_\pi$ for some $\pi \in F$ with $\pi(i) = j$ and define $A = \{k \mid x_{\pi(k)} < x_{\pi(i)} \lor (x_{\pi(k)} = x_{\pi(i)} \land \pi(k) \leq \pi(i))\}$, then one easily verifies that $\Lambda_j(x) = |A|$. We prove that $\Lambda_j(x) = i$ by showing that $A = \{1, \ldots, i\}$. Distinguish the following cases for $k \in \{1, \ldots, n\}$:

- $k = i$ Obviously $k \in A$.
- $k < i$ We show that $k \in A$. Firstly, since $x \in \Omega_\pi$ we have using (15) that $x_{\pi(k)} \leq x_{\pi(k+1)} \leq \cdots \leq x_{\pi(i)}$. If $x_{\pi(k)} = x_{\pi(i)}$ then $x_{\pi(k)} = x_{\pi(k+1)} = \cdots = x_{\pi(i)}$, and hence, using (15) again, $\pi(k) < \pi(k+1) < \cdots < \pi(i)$.
- $k > i$ Copy the lines for the case $k < i$ with $k$ and $i$ interchanged to show that $k \notin A$.

($\supseteq$) Suppose $\Lambda_j(x) = i$, then there exists an $1 \leq s \leq i$ such that $\{k \mid x_k < x_j \lor (x_k = x_j \land k \leq j)\} = \{k_1, \ldots, k_s, \ldots, k_i\}$, satisfying $x_{k_l} < x_j$ for $1 \leq l \leq s$ and $x_{k_l} = x_j$ for $s \leq l \leq i$. Let $\{k \mid x_k > x_j \lor (x_k = x_j \land k > j)\} = \{k_{i+1}, \ldots, k_n\}$, then without loss of generality we assume $x_{k_l} < x_{k_{i+1}} \lor (x_{k_l} = x_{k_{i+1}} \land k_l < k_{i+1})$, for $1 \leq l \leq s$, $s \leq l \leq i$ and $i < l < n$, respectively. Finally, define $\pi \in F$ by $\pi(l) = k_l$, $(l = 1, \ldots, n)$, then the verifications of $\pi(i) = j$ and $x \in \Omega_\pi$ are straightforward. \hfill $\square$

In the following theorem we present a three-layered perceptron that solves the problem of sorting $n$ numbers, using $\frac{1}{2}n(n-1)$ units in the first hidden layer and $n^2$ units in both the second hidden layer and the output layer.

**Theorem 4** Let $f : \mathbb{R}^n \to \{0, 1\}^{n \times n}$ be the function $f = g^{(3)} \circ g^{(2)} \circ g^{(1)}$, with $g^{(l)} = \Theta(\tilde{g}^{(l)})$ and $\tilde{g}^{(l)}$ for $l = 3, 2, 1$ defined by, respectively:

$$\forall i,j=1,\ldots,n,i<j : \tilde{g}^{(1)}_{ij}(x) = \sum_{k=1}^{n} (1_{k=j} - 1_{k=i}) x_k,$$

$$\forall i,j=1,\ldots,n : \tilde{g}^{(2)}_{ij}(w) = \sum_{k=1}^{n} \sum_{l=\max(i+1)}^{n} (1_{l=j} - 1_{k=j}) w_{kl} + n + 1 - i - j, \quad (19)$$

$$\forall i,j=1,\ldots,n : \tilde{g}^{(3)}_{ij}(v) = \sum_{k=1}^{n} \sum_{l=1}^{n} (1_{k=i} - 1_{k=i+1}) 1_{l=j} v_{kl} - 1.$$
Then \( f \) represents a three-layered perceptron that solves \((\Omega, \Gamma)\).

**Proof**

We firstly show that \( A_j(x) \geq i \text{ iff } (\hat{g}_{ij}^{(2)} \circ g^{(1)})(x) \geq 0:\)

\[
A_j(x) = |\{k \mid x_k < x_j \land (x_k = x_j \land k \leq j)\}|
\]

\[
= 1 + |\{k < j \mid x_k \leq x_j\}| + |\{k > j \mid x_k < x_j\}|
\]

\[
= 1 + \sum_{k<j} g_{ij}^{(1)}(x) + \sum_{k>j} (1 - g_{jk}^{(1)}(x))
\]

\[
= (\hat{g}_{ij}^{(2)} \circ g^{(1)})(x) + i.
\]

In combination with Lemma 3 this yields:

\[
V^{(ij)}_1 = \{x \mid A_j(x) = i\}
\]

\[
= \{x \mid A_j(x) \geq i \land A_j(x) < i + 1\}
\]

\[
= \{x \mid (g_{ij}^{(2)} \circ g^{(1)})(x) = 1 \land (\hat{g}_{i+1,j}^{(2)} \circ g^{(1)})(x) = 0\}
\]

\[
= \{x \mid (g_{ij}^{(3)} \circ (g^{(2)} \circ g^{(1)}))(x) = 1\},
\]

which completes the proof. \(\square\)

The three-layered perceptron for the sorting of \( n \) numbers, presented in Theorem 4 has a network configuration that consists of three layers, \( n \) inputs, \( \frac{1}{2}n(n-1) \) units in the first hidden layer, \( n^2 \) units in the second hidden layer and \( n^2 \) outputs. As an example we consider the case \( n = 3 \). Using (19) we find for the first layer:

\[
g_{12}^{(1)}(x) = x_2 - x_1,
\]

\[
g_{13}^{(1)}(x) = x_3 - x_1,
\]

\[
g_{23}^{(1)}(x) = x_3 - x_2,
\]

the second layer:

\[
g_{11}^{(2)}(w) = -w_{12} - w_{13} + 3 - i,
\]

\[
g_{12}^{(2)}(w) = w_{12} - w_{23} + 2 - i,
\]

\[
g_{13}^{(2)}(w) = w_{13} + w_{23} + 1 - i,
\]

and the third layer:

\[
g_{1j}^{(3)}(v) = v_{1j} - v_{2j} - 1,
\]

\[
g_{2j}^{(3)}(v) = v_{2j} - v_{3j} - 1,
\]

\[
g_{3j}^{(3)}(v) = v_{3j} - 1.
\]

In Figure 1 the configuration of the corresponding three-layered perceptron is given.
3

2

1

INPUT $x_1$ $x_2$ $x_3$

OUTPUT

Figure 1: The network configuration of a three-layered perceptron that exactly solves the problem of sorting 3 numbers.

4 Discussion

The structure of the three-layered perceptron given by Theorem 4 is remarkably simple. In fact the given ‘neural’ solution is closely related to the parallel enumerative sorting algorithm of Preparata; see Preparata [5] and Kronsjö [3]. The three layers of the perceptron correspond to the three tasks distinguished in Preparata’s algorithm:

- Count acquisition,
• Rank computation,
• Data rearrangement.

Preparata's algorithm requires $O(n^2)$ processors and completes the sorting in $O(\log n)$ time. If we use a straightforward simulation of the obtained three-layered perceptron on a PRAM, we need $O(n^3)$ processors and $O(\log n)$ time (see [7]). This number of processors is necessary since we have $O(n^3)$ connections between the second and the first layer. However, a large part of the calculations is identical and it can be shown that $O(n^2)$ processors suffice, in which case we obtain Preparata's algorithm. In an idealized situation, all the calculations of one layer of a multi-layered perceptron can be done in constant parallel time. In this situation the time and space complexity are $O(1)$ and $O(n^2)$, respectively. Finally, we note that the found network structure is basically the same as the structure used in the first layers of the 5-layered perceptron used by Chen and Hsieh for solving the sorting problem; see Chen & Hsieh [1].

5 Concluding Remarks

In this paper we have presented two three-layered perceptrons for the problem of sorting $n$ numbers. The first one has been obtained by formulating the sorting problem as a combinatorial optimization problem and applying the general method of finding a multi-layered perceptron that solves given combinatorial optimization problem presented in [7]. By careful examination of the cost function the number of required units could be substantially reduced in two steps. In the first and second step of the reduction the sizes of the first and second hidden layer were reduced. This yields a three-layered perceptron that solves the problem of sorting $n$ numbers using a total of $\frac{5}{2}n(n - 1)$ units. The structure of the final result is easily interpreted and found to be closely related to parallel enumerative sorting algorithm of Preparata [5]. Hence, the results can also be viewed as a mathematical model for that algorithm.

If in general a problem can be solved by a multi-layered perceptron with a fixed number of layers and a polynomial number of units, the problem can be solved on a PRAM in polylogarithmical time, which implies that the problem is in POLYLOGSPACE (see [7]). Since it is unlikely that a $P$-complete problem is in POLYLOGSPACE, it is unlikely to find efficient multi-layered perceptrons for such a problem. On the other hand, if a problem is known to be in POLYLOGSPACE, such as the sorting problem and the shortest path problem, finding an efficient multi-layered perceptron becomes a challenging problem.
References


<table>
<thead>
<tr>
<th>Number</th>
<th>Month</th>
<th>Author</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>92-01</td>
<td>January</td>
<td>F.W. Steutel</td>
<td>On the addition of log-convex functions and sequences</td>
</tr>
<tr>
<td>92-02</td>
<td>January</td>
<td>P. v.d. Laan</td>
<td>Selection constants for Uniform populations</td>
</tr>
<tr>
<td>92-04</td>
<td>February</td>
<td>H.J.C. Huijberts H. Nijmeijer</td>
<td>Strong dynamic input-output decoupling: from linearity to nonlinearity</td>
</tr>
<tr>
<td>92-05</td>
<td>March</td>
<td>S.J.L. v. Eijndhoven J.M. Soethoudt</td>
<td>Introduction to a behavioral approach of continuous-time systems</td>
</tr>
<tr>
<td>92-06</td>
<td>April</td>
<td>P.J. Zwietering E.H.L. Aarts J. Wessels</td>
<td>The minimal number of layers of a perceptron that sorts</td>
</tr>
<tr>
<td>92-07</td>
<td>April</td>
<td>F.P.A. Coolen</td>
<td>Maximum Imprecision Related to Intervals of Measures and Bayesian Inference with Conjugate Imprecise Prior Densities</td>
</tr>
<tr>
<td>92-08</td>
<td>May</td>
<td>I.J.B.F. Adan J. Wessels W.H.M. Zijm</td>
<td>A Note on “The effect of varying routing probability in two parallel queues with dynamic routing under a threshold-type scheduling”</td>
</tr>
<tr>
<td>92-10</td>
<td>May</td>
<td>P. v.d. Laan</td>
<td>Subset Selection: Robustness and Imprecise Selection</td>
</tr>
<tr>
<td>92-11</td>
<td>May</td>
<td>R.J.M. Vaessens E.H.L. Aarts J.K. Lenstra</td>
<td>A Local Search Template (Extended Abstract)</td>
</tr>
<tr>
<td>92-12</td>
<td>May</td>
<td>F.P.A. Coolen</td>
<td>Elicitation of Expert Knowledge and Assessment of Imprecise Prior Densities for Lifetime Distributions</td>
</tr>
<tr>
<td>92-13</td>
<td>May</td>
<td>M.A. Peters A.A. Stoorvogel</td>
<td>Mixed $H_2/H_\infty$ Control in a Stochastic Framework</td>
</tr>
<tr>
<td>Number</td>
<td>Month</td>
<td>Author</td>
<td>Title</td>
</tr>
<tr>
<td>--------</td>
<td>-------</td>
<td>------------------</td>
<td>---------------------------------------------------------</td>
</tr>
<tr>
<td>92-14</td>
<td>June</td>
<td>P.J. Zwietering</td>
<td>The construction of minimal multi-layered perceptrons:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>E.H.L. Aarts</td>
<td>a case study for sorting</td>
</tr>
<tr>
<td></td>
<td></td>
<td>J. Wessels</td>
<td></td>
</tr>
</tbody>
</table>