Analysis and Testing of the m-Class RDP Neural Network

David A. Elizondo, Juan M. Ortiz-de-Lazcano-Lobato, and Ralph Birkenhead

Abstract. The Recursive Deterministic Perceptron (RDP) feed-forward multilayer neural network is a generalisation of the single layer perceptron topology. This model is capable of solving any two-class classification problem unlike the single layer perceptron which can only solve classification problems dealing with linearly separable sets. For all classification problems, the construction of an RDP is done automatically and convergence is always guaranteed. A generalisation of the 2-class Recursive Deterministic Perceptron (RDP) exists. This generalisation always allows the deterministic separation of m-classes. It is based on a new notion of linear separability and it arises naturally from the 2 valued RDP. The methods for building 2-class RDP neural networks have been extensively tested. However, no testing has been done before on the m-class RDP method. For the first time, a study on the performance of the m-class method is presented. This study will allow the highlighting of the main advantages and disadvantages of this method by comparing the results obtained while building m-class RDP neural networks with other more classical methods such as Backpropagation and Cascade Correlation in terms of level of generalisation and topology size. The networks were trained and tested using the following standard benchmark classification datasets: Glass, Wine, Zoo, Iris, Soybean, and Wisconsin Breast Cancer.

1 Introduction

The RDP for 2-class classification problems was introduced in [12]. This topology is a generalisation of the single layer perceptron topology (SLPT) developed

David A. Elizondo and Ralph Birkenhead
School of Computing, De Montfort University, The Gateway, Leicester,
LE1 9BH, United Kingdom
e-mail: {elizondo,rab}@dmu.ac.uk

Juan M. Ortiz-de-Lazcano-Lobato
School of Computing, University of Málaga, Bulevar Louis Pasteur, 35, Málaga, Spain
e-mail: jmortiz@lcc.uma.es

L. Franco et al. (Eds.): Constructive Neural Networks, SCI 258, pp. 171–192.
springerlink.com

by Rosenblatt [11]. This generalisation is capable of transforming any non-linearly separable (NLS) 2-class classification problem into a linearly separable (LS) one, thus making it possible for the SLPT to find a solution to the problem. An extension of the RDP algorithm to m-class problems (with $m \geq 2$) was introduced in [13]. This extension is based on a new notion of linear separability and, it evolves naturally from the 2-valued RDP.

1.1 Preliminaries

We use the following standard notions:
• $\mathcal{S}_m$ stands for the set of permutations of $\{1, ..., m\}$.
• If $\mathbf{u} = (u_1, ..., u_d), \mathbf{v} = (v_1, ..., v_d) \in I\!R^d$, then $\mathbf{u}^T\mathbf{v}$ stands for $u_1v_1 + ... + u_dv_d$; and $\mathbf{u}(j) = u_j$ (i.e. $\mathbf{u}(j)$ is the *j-th* component of $\mathbf{u}$).
• $\Pi_{\{i_1,...,i_k\}}\mathbf{u} = (u_{i_1}, ..., u_{i_k})$ and by extension,
if $S \subset I\!R^d$ then $\Pi_{\{i_1,...,i_k\}}S = \{\Pi_{\{i_1,...,i_k\}}\ \mathbf{x} \mid \mathbf{x} \in S\}$.
• Let $r \in I\!R$, $Adj(\mathbf{u}, r) = (u_1, ..., u_d, r)$ and by extension,
if $S \subset I\!R^d,\ Adj(S, r) = \{Adj(\mathbf{x}, r) \mid \mathbf{x} \in S\}$.
• $Im(E, F) = \{(x_1, ..., x_d, x_{d+1}) \in F \mid (x_1, ..., x_d) \in E\}$ is defined for $E \subset I\!R^d$ and $F \subset I\!R^{d+1}$.
• $\mathcal{P}(\mathbf{w}, t)$ stands for the hyperplane $\{\mathbf{x} \in I\!R^d \mid \mathbf{w}^T\mathbf{x} + t = 0\}$ of $I\!R^d$.

1.2 Some Definitions and Properties

In this section, we introduce the notions of convex hull(CH), [10], and of linear separability. A discussion on the different methods for testing linear separability can be found in [12].

Definition 1. Let S be a sub-set of $I\!R^d$,$CH(S) = \{t_1\mathbf{x}_1 + \ldots + t_k\mathbf{x}_k \mid \mathbf{x}_1, \ldots, \mathbf{x}_k \in S,$ $t_1, \ldots, t_k \in [0, 1]$ and $t_1 + \ldots + t_k = 1\}$.
Thus, if S is finite, then there exists $\mathbf{a}_1, ..., \mathbf{a}_k \in I\!R^d$ and $b_1, ..., b_k \in I\!R$ such that $CH(S) = \{\mathbf{x} \in I\!R^d \mid \mathbf{a}_i^T\mathbf{x} \geq b_i\ for\ 1 \leq i \leq k\}$.

Definition 2. Two subsets X and Y of $I\!R^d$ are said to be linearly separable if there exists a hyperplane $\mathcal{P}(\mathbf{w}, t)$ of $I\!R^d$, such that ($\forall \mathbf{x} \in X,\ \mathbf{w}^T\mathbf{x} + t > 0$ and $\forall \mathbf{y} \in Y,\ \mathbf{w}^T\mathbf{y} + t < 0$). In the following we will denote the fact that X and Y are LS by $X \mid\mid Y$ or $X \mid\mid Y\ (\mathcal{P}(\mathbf{w}, t))$ if we want to specify the hyperplane which linearly separates X and Y.

This paper is divided into four sections. The m-class generalisation of the RDP neural network, based on a notion of linear separability for m classes, is presented in section two. In this section also, some of the notions used throughout this paper are introduced. In section three, the procedure used to evaluate the generalisation of the m-class RDP model is presented. Six machine learning benchmarks (Iris, Soybean, and Wisconsin Breast Cancer) were used [3] and datasets were generated using cross validation. The method is compared with Backpropagation and Cascade Correlation in terms of their level of generalisation. A summary and some conclusions are presented in section four.

2 The m-Class RDP Algorithm

The m-class RDP algorithm is an adaptation of the 2-class RDP based on the following notion of linear separability for m classes ($m > 2$).

Definition 3. Let $X_1, ..., X_m \subset I\!R^d$ and $a_0 < a_1 < ... < a_m$, $X_1, ..., X_m$ are said to be linearly separable relatively to the ascending sequence of real numbers $a_0, ..., a_m$ if
$\exists \sigma \in \mathcal{S}_m, \exists \mathbf{w} \in I\!R^d, \exists t \in I\!R \ \ such \ \ that \ \ \forall i, \forall \mathbf{x} \in X_{\sigma(i)}, \ a_{i-1} < \mathbf{w}^T\mathbf{x} + t < a_i.$

Remarks
Let $X_1, ..., X_m \subset I\!R^d$ and $a_0 < a_1 < a_2 < ... < a_m$,
- $X_1, ..., X_m$ are linearly separable relatively to $a_0, ..., a_m$ iff $CH(X_1), ..., CH(X_m)$ are linearly separable relatively to $a_0, ..., a_m$.
- Let $\sigma \in \mathcal{S}_m$.
Put: $X^\sigma = Adj(X_{\sigma(1)}, -a_0) \cup Adj(X_{\sigma(2)}, -a_1) ... \cup Adj(X_{\sigma(m)}, -a_{m-1})$, $Y^\sigma = Adj(X_{\sigma(1)}, -a_1) \cup Adj(X_{\sigma(2)}, -a_2) ... \cup Adj(X_{\sigma(m)}, -a_m)$, then, $X_1, ...,$ X_m are linearly separable relatively to $a_0, ..., a_m$ by using σ iff $X^\sigma \ || \ Y^\sigma$. In other words, we reduce the problem of linear separability for m classes to the problem of linear separability for 2 classes. We do this by augmenting the dimension of the input vectors with the ascending sequence $a_0, ..., a_m$.
- If $X_1 \ || \ X_2$ $(\mathcal{P}(\mathbf{w}, t))$ and $\alpha = Max(\{|\mathbf{w}^T\mathbf{x} + t| \ ; \ \mathbf{x} \in (X_1 \cup X_2)\}$, then X_1, X_2 are linearly separable relatively to $-\alpha, 0, \alpha$.

Proposition 1. *Let* $X_1, ..., X_m \subset I\!R^d$*,* $a, b \in I\!R, h, k > 0$ *and let* $a_i = a + ih, b_i = b + ik$*, for* $0 \leq i \leq m$*, then* $X_1, ..., X_m$ *are linearly separable relatively to* $a_0, ..., a_m$ *iff they are linearly separable relatively to* $b_0, ..., b_m$*. In other words, the linear separability between* m *classes is independent of the arithmetic sequence.*

Proof. Let $\sigma \in \mathcal{S}_m$ represent a class, and let $\mathbf{w} \in I\!R^d, t \in I\!R$ such that $\forall i, \forall \mathbf{x} \in X_{\sigma(i)}, \ a_{i-1} < \mathbf{w}^T\mathbf{x} + t < a_i$.

Thus, $\forall i, \forall \mathbf{x} \in X_{\sigma(i)}, \ b_{i-1} < \frac{k}{h}\mathbf{w}^T\mathbf{x} + \frac{k}{h}(t - a) + b < b_i$ □

Definition 4. $X_1, ..., X_m \subset I\!R^d$ are said to be linearly separable if there exists $a \in I\!R, h > 0$ such that $X_1, ..., X_m$ are linearly separable relatively to $a, a + h, ...,$ $a + mh$.

Definition 5. A m-SLPT with the weight $\mathbf{w} \in I\!R^d$, the threshold $t \in I\!R$, the values $v_1, v_2, ..., v_m \in I\!R$ and the characteristic $(c, h) \in I\!R \times I\!R^+$ (*c represents the value corresponding to the starting hyperplane, and h a chosen distance between a hyperplane which we will call the step size*), has the same topology as the 2-class SLPT.

The only difference is that the function corresponding to a m-SLPT is a $m-$valued function f defined by : $\forall \mathbf{y} \in \mathbb{R}^d$

$$f(\mathbf{y}) = \begin{cases} v_1 & \text{if} \quad \mathbf{w}^T\mathbf{y} + t < c + h \\ v_i & \text{if} \quad c + (i-1)h < \mathbf{w}^T\mathbf{y} + t < c + ih, \text{ for } 1 < i < m \\ v_m & \text{if} \quad \mathbf{w}^T\mathbf{y} + t > c + (m-1)h \end{cases} \tag{1}$$

2.1 The Specialised NLS to LS Transformation Algorithm for m Classes

A specialised version of the transformation algorithm , from two to m classes, was proposed in [12]. This extension is based on the notion of linear separability for m classes described above.

Let $c \in \mathbb{R}, h > 0, m$ be the number of classes and $b = -(m - \frac{3}{2})h$,for $1 \leq i < m$ $b_i = c + (m-i)b + (\frac{(m-1)(m-2)}{2} - \frac{(i-1)(i-2)}{2})h$ and $b_m = c$.

Table 1 shows the specialized NLS to LS transformation algorithm for m classes. We proceed as in the 2-class specialized transformation algorithm. That is to say, at each step we select a LS sub-set which belongs to a single class and add an artificial variable to the entire input data set. To this artificial variable we assign a value b_i for all the input vectors belonging to the selected LS sub-set and a value b_j to the rest of the set of input vectors, where $b_i \neq b_j$. Two cases for assigning the values to the artificial inputs are possible depending on the class to which the LS sub-set belongs:

1. If the selected LS sub-set belongs to the jth class, with $j < m$, we add to its input vector a new component with value b_j and we add to the rest of the input vector a new component with value b_{j+1}.
2. If the selected LS sub-set belongs to the last class *(mth class)*, we add to its input vector a new component with value b_m and we add to the rest of the input vector a new component with value b_{m-1}.

In the following theorem we prove the correctness and the termination of the algorithm presented in table 1 which allows the construction of an $m-$RDP for linearly separating any given m classes.

Theorem 1. *If $X_1^i, ..., X_m^i$ are not linearly separable, then there exists Z_i such that ($Z_i \subset {X'_1}^i$ or ... or $Z_i \subset {X'_m}^i$), $Z_i \neq \emptyset$ and $Z_i \mid\mid (S_i \ \setminus \ Z_i)$.*

Proof. We will prove that, there exists $\mathbf{x} \in {X'_1}^i \cup ... \cup {X'_m}^i$ such that $\{\mathbf{x}\} \mid\mid (S_i \ \setminus \ \{\mathbf{x}\})$.

Assume that $\forall \mathbf{x} \in {X'_1}^i \cup ... \cup {X'_m}^i, \quad \{\mathbf{x}\}$ and $(S_i \ \setminus \ \{\mathbf{x}\})$ are not linearly separable, then ${X'_1}^i \cup ... \cup {X'_m}^i \subset CH(S_i \setminus ({X'_1}^i \cup ... \cup {X'_m}^i))$.

Table 1 Specialized NLS to LS transformation algorithm for m classes.

```
SNLS2LS($X_1, .., X_m, X_{0i}, .., X_{mi}$)
– data: $m$ data set vectors, $X_0, .., X_m$ representing $m$ NLS classes
– result: A $m$-RDP $[(\mathbf{w}_0, t_0, a_0, h_0, b_{0,1}, b_{0,2})$, ..., $(\mathbf{w}_{i-1}, t_{i-1}, a_{i-1}, h_{i-1}, b_{i-1,1}, b_{i-1,2})$,
$(\mathbf{w}_i, t_i, a_i, h_i, b_1, ..., b_m)]$ which linearly separates $X_1, ..., X_m$.
INITIALIZE : Let $i := 0$;  $X_1^0 := X_1; ...; X_m^0 := X_m$;  ${X_1'}^0 := X_1; ...; {X_m'}^0 := X_m$;
$S_0 = X_1 \cup ... \cup X_m$;
    WHILE  $(X_1^i, ..., X_m^i)$ are not linearly separable
    BEGIN
        SELECT : Select a non-empty sub-set $Z_i$ from ${X_1'}^i$ or ... or from ${X_m'}^i$
            (if it exists) such that $Z_i$, $(S_i \setminus Z_i)$ are linearly separable
            (i.e. ($Z_i \subset {X_1'}^i$ or ... or $Z_i \subset {X_m'}^i$) and $Z_i \parallel (S_i \setminus Z_i)$) $(\mathcal{P}(\mathbf{w}_i, t_i))$ ;
            CASE :
                    Case $Z_i \subset {X_1'}^i$ :
                            $S_{i+1} := Adj(Z_i, b_1) \cup Adj(S_i \setminus Z_i, b_2)$;
                            ${X_1'}^{i+1} := Im({X_1'}^i, S_{i+1}) \setminus Im(Z_i, S_{i+1})$;
                            ${X_2'}^{i+1} := Im({X_2'}^i, S_{i+1}); ...;$
                            ${X_2'}^{i+1} := Im({X_m'}^i, S_{i+1})$;
                            $X_1^{i+1} := Im(X_1^i, S_{i+1})$;
                            $X_2^{i+1} := Im(X_2^i, S_{i+1}); ...;$
                            $X_m^{i+1} := Im(X_m^i, S_{i+1})$;
                            $i := i + 1$;
                            .........
                    Case  $Z_i \subset {X_j'}^i$ :
                            $S_{i+1} := Adj(Z_i, b_j) \cup Adj(S_i \setminus Z_i, b_{j+1})$;
                            ${X_1'}^{i+1} := Im({X_1'}^i, S_{i+1}); ...;$
                            ${X_{j-1}'}^{i+1} := Im({X_{j-1}'}^i, S_{i+1})$;
                            ${X_j'}^{i+1} := Im({X_j'}^i, S_{i+1}) \setminus Im(Z_i, S_{i+1})$;
                            ${X_{j+1}'}^{i+1} := Im({X_{j+1}'}^i, S_{i+1}); ...;$
                            ${X_m'}^{i+1} := Im({X_m'}^i, S_{i+1})$;
                            $X_1^{i+1} := Im(X_1^i, S_{i+1})$;
                            $X_2^{i+1} := Im(X_2^i, S_{i+1}); ...;$
                            $X_m^{i+1} := Im(X_m^i, S_{i+1})$;
                            $i := i + 1$;
                            .........
                    Case $Z_i \subset {X_m'}^i$ :
                            $S_{i+1} := Adj(Z_i, b_m) \cup Adj(S_i \setminus Z_i, b_{m-1})$;
                            ${X_1'}^{i+1} := Im({X_1'}^i, S_{i+1}); ...;$
                            ${X_{m-1}'}^{i+1} := Im({X_{m-1}'}^i, S_{i+1})$;
                            ${X_m'}^{i+1} := Im({X_m'}^i, S_{i+1}) \setminus Im(Z_i, S_{i+1})$;
                            $X_1^{i+1} := Im(X_1^i, S_{i+1})$;
                            $X_2^{i+1} := Im(X_2^i, S_{i+1}); ...;$
                            $X_m^{i+1} := Im(X_m^i, S_{i+1})$;
                            $i := i + 1$;
    END
```

So, if $S_i = \{\mathbf{v}_1, \ldots, \mathbf{v}_k, \mathbf{v}_{k+1}, \ldots, \mathbf{v}_{k+r_1}, \mathbf{v}_{k+r_1+1}, \ldots, \mathbf{v}_{k+r_1+r_2}, \ldots, \mathbf{v}_{k+r_1+\ldots+r_m}\}$

where $X_1'^{\,i} \cup ... \cup X_m'^{\,i} = \{\mathbf{v}_1, ..., \mathbf{v}_k\}$ and

$X_j^i \setminus X_j'^{\,i} = \{\mathbf{v}_{k+r_1+...+r_{j-1}+1}, ..., \mathbf{v}_{k+r_1+...+r_j}\}$ for $1 \leq j \leq m$.

Let $\mathbf{x} \in X_1'^{\,i} \cup ... \cup X_m'^{\,i}$, then $\mathbf{x} = t_1\mathbf{v}_{k+1} + ... + t_{r_1+...+r_m}\mathbf{v}_{k+r_1+...+r_m}$, $t_1, ..., t_{r_1+...+r_m} \geq 0$ and $t_1 + ... + t_{r_1+...+r_m} = 1$.

Let $j < m$, $1 \leq l \leq r_j$ and e_l such that $\mathbf{v}_{k+r_1+...+r_{j-1}+l}(e_l) = b_j$ and $\mathbf{v}_f(e_l) = b_{j+1}$ for $\mathbf{v}_f \notin X_j^i \setminus X_j'^{\,i}$ ($b_j < b_{j+1}$, $\mathbf{x}(e_l) = b_{j+1}$) . If $t_{r_1+...+r_{j-1}+l} > 0$ then $b_{j+1} = \mathbf{x}(e_j) < (t_1 + ... + t_{k+r_1+...+r_m})b_{j+1} = b_{j+1}$, which is absurd; thus, $\forall j \leq r_1 + ... + r_{m-1},\ t_j = 0$.

Let $j < r_m$ and $1 \leq l \leq i$ such that $\mathbf{v}_{k+r_1+...+r_{m-1}+j}(l) = b_m$ and $\mathbf{v}_f(l) = b_{m-1}$ for $\mathbf{v}_f \notin X_m^i \setminus X_m'^{\,i}$ $(b_{m-1} < b_m$, $\mathbf{x}(l) = b_{m-1})$. If $t_{r_1+...+r_{m-1}+j} > 0$ then $b_{m-1} = \mathbf{x}(l) > (t_{r_1+...+r_{m-1}+1} + ... + t_{r_1+...+r_{m-1}+r_m})b_{m-1} = b_{m-1}$, which is absurd; then, $\forall j \leq r_m,\ t_{r_1+...+r_{m-1}+j} = 0$. Thus $\mathbf{x} \notin CH((S_i \setminus (X_1'^{\,i} \cup ... \cup X_m'^{\,i}))$.

So, there exists $\mathbf{x} \in X_1'^{\,i} \cup ... \cup X_m'^{\,i}$ such that $\{\mathbf{x}\} \mid\mid (S_i \setminus \{\mathbf{x}\})$ □

Theorem 2. *If $X_1'^{\,i} \cup ... \cup X_m'^{\,i} = \emptyset$ then $X_1'^{\,i}, ..., X_m'^{\,i}$ are linearly separable. Thus, the algorithm stops, in the worse case, after Card$(X_1 \cup \ldots \cup X_m) - 1$ steps, and the result $[(\mathbf{w}_0, t_0,\ a_0, h_0,\ b_{0,1}, b_{0,2}),\ ..., (\mathbf{w}_{i-1},\ t_{i-1}, a_{i-1},\ h_{i-1}, b_{i-1,1},\ b_{i-1,2}), (\mathbf{w}_i, t_i,\ a_i, h_i,\ b_1,\ ..., b_m)]$ is a m-RDP separating $X_1, ..., X_m$, where $b_{j,1}, b_{j,2}$ are b_k, b_{k+1} if at step j, $Z_j \subset X_k'^{\,j}$ and $k \leq m-1$ or b_{m-1}, b_m if at step j, $Z_j \subset X_m'^{\,j}$.*

Proof. Let $\mathbf{w} = (0, ..., 0, 1, ..., 1) \in I\!R^{d+i}$ with d times 0 and i times 1, and let $t = -(k_{X_1}(i)b_2 + ... + k_{X_{m-1}}(i)b_m + k_{X_m}(i)b_{m-1})$. Thus, $\forall j < m, \forall \mathbf{x} \in X_j^i,\quad \mathbf{w}^T\mathbf{x} + t = b_j - b_{j+1} = b + (j-1)h$, and $\forall \mathbf{x} \in X_m^i,\quad \mathbf{w}^T\mathbf{x} + t = b_m - b_{m-1} = b + (m-1)h$. Let $a_0 = b - \frac{h}{2}, a_i = a_0 + ih$ for $1 \leq i \leq m$, thus $\forall j \leq m,\ \forall \mathbf{x} \in X_j^i,\quad a_{j-1} \leq \mathbf{w}^T\mathbf{x} + t \leq a_j$.

So, $X_1^i, ..., X_m^i$ are linearly separable by the hyperplane $\mathcal{P}(\mathbf{w}, t)$ □

3 Comparison Procedure

The six machine learning benchmark data sets used in the comparison study identified in section 1 are described briefly.

The Glass benchmark relates to the classification of types of glass on criminological investigation. The glass found at the scene of a crime can be used as evidence. This benchmark consists of nine inputs and six classes (2). The dataset contains a total of 214 samples.

The Wine dataset contains results of a chemical analysis of wines grown in the same region in Italy derived from three different crops. The analysis determined the quantities of 13 constituents found in each of the three types of wines (3).

Table 2 Inputs and outputs used in the Glass classification problem.

Attributes (In cm)	Output	Output Classes
RI: refractive index	Type of Glass	building windows float processed
Na: Sodium		building windows non float processed
Mg: Magnesium		vehicle windows float processed
Al: Aluminium		containers
Si: Silicon		tableware
K: Potassium		headlamps
Ca: Calcium		
Ba: Barium		
Fe: Iron		

Table 3 Inputs and outputs used in the Wine classification problem.

Attributes (1 - 13)	Output	Output Classes
Alcohol	Class	1
Malic acid		2
Ash		3
Alcalinity of ash		
Magnesium		
Total phenols		
Flavanoids		
Nonflavanoid phenols		
Proanthocyanins		
Colour intensity		
Hue		
OD280/OD315 of diluted wines		
Proline		

Table 4 Inputs and outputs used in the Soybean classification problem.

Attributes		Output	Output classes
Date	leaf-shred	Disease type	brown-spot
plant-stand	stem		alternarialeaf-spot
precipitation	stem-cankers		frog-eye-leaf-spot
temperature	canker-lesion		
hail	fruiting-bodies		
crop-hist	external decay		
area-damaged	fruit-pods		
severity	fruit spots		
seed-tmt	seed		
germination	plant-growth		

Table 5 Inputs and outputs used in the Zoo classification problem.

Attributes (1 - 17)	Output	Output Classes
animal name	Animal	1 (41 samples of mammals)
hair		2 (20 samples of birds)
feathers		3 (5 samples of reptiles)
eggs		4 (13 samples of fish)
milk		5 (4 samples of frogs)
airborne		6 (8 samples of insects)
aquatic		7 (10 samples of sea shell)
predator		
toothed		
backbone		
breathes		
venomous		
fins		
legs		
tail		
domestic		
catsize		
type		

Table 6 Inputs and outputs used in the Iris classification problem.

Attributes (In cm)	Output	Output Classes
Sepal Length	Iris plant type	Iris Setosa
Sepal Width		Iris Versicolour
Petal Length		Iris Virginica
Petal Width		

Table 7 Inputs and outputs used in the Wisconsin Breast Cancer classification problem.

Attributes (1 - 10)	Output	Output Classes
Clump Thickness	Class	Benign
Uniformity of Cell Size		Malignant
Uniformity of Cell Shape		
Marginal Adhesion		
Single Epithelial Cell Size		
Bare Nuclei		
Bland Chromatin		
Normal Nucleoli		
Mitoses		

The Zoo benchmark data set contains 17 Boolean-valued attributes with a type of animal as output (5). A total of 101 samples are included (mammals, birds, reptiles, fish, frogs, insects and sea shells).

The Iris dataset classifies a plant as being an Iris Setosa, Iris Versicolour or Iris Virginica. The dataset describes every iris plant using four input parameters (Table 6). The dataset contains a total of 150 samples with 50 samples for each of the three classes. All the samples of the Iris Setosa class are linearly separable from the rest of the samples (Iris Versicolour and Iris Virginica). Some of the publications that used this benchmark include: [7] [8] [2] and [4].

The Soybean classification problem contains data for the disease diagnosis of the Soybean crop. The dataset describes the different diseases using symptoms. The original dataset contains 19 diseases and 35 attributes. The attribute list was limited to those attributes that had non trivial values in them (Table 4). Thus there were only 20 out of the 35 attributes included in the tests. Only 15 of the 19 have no missing values. Therefore, only these 15 classes were used for the comparisons.

The Wisconsin Breast Cancer dataset [9, 1, 15] consists of a binary classification problem to distinguish between benign and malignant breast cancer. The data set contains 699 instances and 9 attributes (Table 7). The class distribution is: Benign 458 instances (65.5 %), and Malignant 241 instances (34.5 %).

The technique of cross validation was applied to split the benchmarks into training and testing data sets. The datasets were randomly divided into 'n' equal sized testing sets that were mutually exclusive [14]. The remaining samples were used to train the networks. In this study, the classification benchmark data sets were divided into ten equally sized data sets. On one hand sixty percent of the samples were used for training the networks and the remaining forty percent were used for testing purposes. On the other hand the training dataset consisted of eighty percent of the samples and the remaining twenty percent were used for the testing dataset.

The simplex algorithm was used on this study for testing for linear separability. This algorithm was remarkably faster than the Perceptron one when searching for LS subsets. Other algorithms for testing linear separability include the Class of Linear Separability [5] and the Fisher method (see [6] for a survey on methods for testing linear separability).

These results provide a good basis for further developing this study and comparing the effects of using single or multiple output neurons for multiple class classification problems using the m-class RDP method and Backpropagation and Cascade Correlation. After describing the experimental setup, some conclusions are presented in the next section.

4 Results and Discussion

We now present a comparison of the m-class RDP construction method, Backpropagation and Cascade Correlation based on their level of generalisation on previously

Table 8 Results obtained with the m-class, and backpropagation, using the Glass data set benchmark in terms of the level of generalisation with 60% of the data used for training and 40% for testing.

Data Set	m-RDP	BackProp	BP 1 out	CC Mout
1	48.84	56.98	52.33	53.49
2	52.33	61.63	53.49	60.47
3	53.49	61.63	59.30	56.98
4	52.33	65.12	55.81	54.65
5	51.16	65.12	50.00	62.79
6	66.28	62.79	55.81	58.14
7	63.95	72.09	51.16	59.30
8	52.33	55.81	59.30	52.33
9	59.30	56.98	61.63	48.84
10	55.81	62.79	52.33	55.81
Mean	55.58	62.01	55.12	56.28
Std	5.76	4.84	3.92	4.14

Table 9 Results obtained with the m-class, and backpropagation, using the Glass data set benchmark in terms of the level of generalisation with 80% of the data used for training and 20% for testing.

Data Set	m-RDP	BackProp	BP 1 out	CC Mout
1	67.44	67.44	67.44	51.16
2	65.12	65.12	55.81	62.79
3	53.49	72.09	55.81	58.14
4	60.47	62.79	60.47	48.84
5	44.19	76.74	62.79	65.12
6	58.14	76.74	65.12	67.44
7	67.44	62.79	51.16	58.14
8	53.49	58.14	51.16	55.81
9	58.14	55.81	55.81	65.12
10	46.51	74.42	67.44	51.16
Mean	57.44	67.21	59.30	58.37
Std	8.13	7.55	6.23	6.62

unseen data and the number of neurons needed for each method to solve the classification problems (i.e. the size of the topology).

As specified before, the m-class RDP uses a single output neuron for multiple classes. Backpropagation and Cascade Correlation are tested using two different topologies. The first one uses a unique output neuron and is named *BP 1out* and *CC 1out* in the tables. The second type of topology uses as many neurons in the output layer as the number of classes in the data set (*Backprop* and *CC Mout* in the tables). Only the first type of topology is used when the dataset defines a binary classification problem such as the Wisconsin Breast Cancer dataset.

Table 10 Results obtained with the m-class, and backpropagation, using the Wine data set benchmark in terms of the level of generalisation with 60% of the data used for training and 40% for testing.

Data Set	m-RDP	BackProp	BP 1out	CC Mout
1	93.06	95.83	94.44	90.28
2	93.06	95.83	98.61	93.06
3	88.89	97.22	95.83	90.29
4	87.50	97.22	94.44	90.28
5	94.44	97.22	95.83	86.11
6	91.67	97.22	93.06	93.06
7	90.28	97.22	94.44	91.67
8	91.67	98.61	95.83	91.67
9	95.83	95.83	94.44	90.28
10	95.83	94.44	95.83	91.67
Mean	92.22	96.66	95.27	90.84
Std	2.79	1.17	1.49	1.99

Table 11 Results obtained with the m-class, and backpropagation, using the Wine data set benchmark in terms of the level of generalisation with 80% of the data used for training and 20% for testing.

Data Set	m-RDP	BP 1out	CC Mout
1	88.89	100.00	88.89
2	86.11	97.22	91.67
3	86.11	97.22	91.67
4	94.44	100.00	94.44
5	94.44	94.44	94.44
6	97.22	100.00	94.44
7	97.22	100.00	97.22
8	97.22	100.00	100.0
9	97.22	97.22	94.44
10	94.44	97.22	91.67
Mean	88.57	98.33	93.88
Std	7.84	1.94	3.15

Overall, considering all the results obtained from tables 8 to 19 in terms of generalisation obtained using the m-class RDP, it appears that the method is broadly comparable with CC and BP, but has slightly poorer results. It appears to be more variable in its performance. While it does generally perform better than the other methods when they are used with a single output neuron, it is arguable that the nature of the data makes this an inappropriate choice of topology for a BP or CC network.

Considering the size of the network produced (tables 20 to 31), the number of neurons in an m-class RDP is usually significantly lower than in the corresponding

Table 12 Results obtained with the m-class, and backpropagation, using the Zoo data set benchmark in terms of the level of generalisation with 60% of the data used for training and 40% for testing.

Data Set	m-RDP	BackProp	BP 1out	CC Mout	CC 1out
1	90.24	87.80	90.24	78.05	97.56
2	92.68	87.80	90.24	92.68	97.56
3	95.12	95.12	95.12	90.24	100.00
4	87.80	97.56	95.12	95.12	97.56
5	92.68	100.00	95.12	92.68	100.00
6	90.24	100.00	97.56	92.68	100.00
7	90.24	100.00	97.56	97.56	100.00
8	92.68	100.00	97.56	97.56	100.00
9	90.24	95.12	100.00	90.24	95.12
10	85.37	90.24	97.56	82.93	92.68
Mean	90.73	95.36	95.60	90.97	98.04
Std	2.77	5.07	3.21	6.20	2.52

Table 13 Results obtained with the m-class, and backpropagation, using the Zoo data set benchmark in terms of the level of generalisation with 80% of the data used for training and 20% for testing.

Data Set	m-RDP	BackProp	BP 1out	CC Mout	CC 1out
1	80.95	100.00	100.00	95.24	100.00
2	80.95	100.00	100.00	95.24	100.00
3	95.24	100.00	100.00	95.24	100.00
4	95.24	95.24	90.48	95.24	100.00
5	85.71	100.00	100.00	95.24	90.48
6	90.48	100.00	100.00	100.00	90.48
7	95.24	95.24	100.00	95.24	100.00
8	76.19	95.24	100.00	85.71	100.00
9	85.71	95.24	100.00	85.71	100.00
10	100.00	100.00	100.00	100.00	100.00
Mean	88.57	98.10	99.05	94.28	98.10
Std	7.84	2.46	3.01	4.92	4.01

BP and CC networks with multiple output neurons. The single output neuron BP and CC networks sometimes have fewer neurons but, as discussed above, this is probably an inappropriate architecture for the data. This will lead to future research and exploring a multiple output architecture for the m-class RDP model.

Table 14 Results obtained with the m-class, and backpropagation, using the Iris data set benchmark in terms of the level of generalisation with 60% of the data used for training and 40% for testing.

Data Set	m-RDP	BackProp	BP 1out	CC Mout	CC 1out
1	98.33	98.33	98.33	98.33	98.33
2	98.33	98.33	98.33	96.67	98.33
3	95.00	96.67	96.67	95	96.67
4	95.00	98.33	98.33	96.67	100.00
5	91.67	98.33	98.33	95	98.33
6	91.67	96.67	96.67	91.67	98.3
3 7	96.67	98.33	96.67	93.33	96.67
8	96.67	96.67	98.33	95	100.00
9	95.00	96.67	96.67	93.33	98.33
10	93.33	91.67	96.67	91.67	96.6
Mean	95.167	97	97.5	94.667	98.166
Std	2.41	2.05	0.87	2.19	1.23

Table 15 Results obtained with the m-class, and backpropagation, using the Iris data set benchmark in terms of the level of generalisation with 80% of the data used for training and 20% for testing.

Data Set	m-RDP	BackProp	BP 1out	CC Mout	CC 1out
1	90.00	93.33	96.67	93.33	91.67
2	90.00	96.67	96.67	93.33	96.67
3	93.33	100.00	100.00	100.00	100.00
4	100.00	100.00	100.00	100.00	96.67
5	96.67	100.00	100.00	100.00	93.33
6	93.33	96.67	100.00	100.00	100.00
7	100.00	96.67	100.00	96.67	96.67
8	96.67	100.00	100.00	100.00	100.00
9	96.67	96.67	96.67	96.67	96.67
10	90.00	93.33	90.00	90.00	93.33
Mean	94.67	97.33	98.00	97	96.50
Std	3.91	2.63	3.22	3.67	2.98

Table 16 Results obtained with the m-class, and backpropagation, using the Soybean data set benchmark in terms of the level of generalisation with 60% of the data used for training and 40% for testing.

Data Set	m-RDP	BackProp	BP 1out	CC Mout	CC 1out
1	83.18	87.85	49.53	87.85	54.21
2	73.83	87.85	54.21	89.71	50.47
3	74.77	84.11	45.79	85.98	47.66
4	61.68	80.37	37.38	83.17	54.21
5	75.70	84.11	45.79	84.11	51.40
6	70.09	81.31	39.25	85.05	51.40
7	75.70	84.11	43.93	85.50	52.34
8	74.77	78.50	47.66	82.24	52.34
9	73.83	84.11	45.79	85.98	52.34
10	71.03	85.05	48.60	91.58	57.94
Mean	73.46	83.74	45.79	86.12	52.43
Std	5.42	3.0	4.87	2.89	2.70

Table 17 Results obtained with the m-class, and backpropagation, using the Soybean data set benchmark in terms of the level of generalisation with 80% of the data used for training and 20% for testing.

Data Set	m-RDP	BackProp	BP 1out	CC Mout	CC 1out
1	74.07	88.89	38.89	83.33	51.85
2	62.96	87.04	40.74	81.48	38.89
3	77.78	92.59	44.44	85.19	64.81
4	87.04	90.74	61.11	92.59	61.11
5	74.07	85.19	42.59	81.48	33.33
6	72.22	90.74	50.00	85.19	50.00
7	83.33	90.74	53.70	87.04	50.00
8	77.78	94.44	55.56	88.89	61.11
9	68.52	92.59	48.15	88.89	55.56
10	74.07	87.04	48.15	88.89	50.00
Mean	75.18	90	48.33	86.30	51.66
Std	6.90	2.92	7.01	3.62	9.85

Table 18 Results obtained with the m-class, and backpropagation, using the Wisconsin Breast Cancer data set benchmark in terms of the level of generalisation with 60% of the data used for training and 40% for testing.

Data Set	m-RDP	BackProp	CC
1	94.16	97.08	98.17
2	95.62	95.26	97.08
3	94.53	95.26	96.00
4	95.62	96.72	97.00
5	93.07	98.18	96.35
6	91.61	97.45	97.00
7	94.16	95.62	96.70
8	89.78	97.45	97.00
9	91.61	97.45	97.45
10	94.53	97.45	98.54
Mean	93.47	96.79	97.13
Std	1.92	1.04	0.77

Table 19 Results obtained with the m-class, and backpropagation, using the Wisconsin Breast Cancer data set benchmark in terms of the level of generalisation with 80% of the data used for training and 20% for testing.

Data Set	m-RDP	BackProp	CC
1	97.08	95.62	93.43
2	94.16	97.81	94.89
3	94.89	97.81	97.08
4	94.16	98.54	93.43
5	95.62	97.81	97.08
6	94.89	96.35	94.16
7	94.16	97.08	94.16
8	92.70	95.62	93.20
9	93.43	96.35	94.16
10	93.43	96.35	93.43
Mean	94.45	96.93	94.65
Std	1.25	1.02	1.46

Table 20 Results obtained with the m-class, and backpropagation, using the Glass data set benchmark in terms of the topology size (number of hidden/intermediate neurons) with 60% of the data used for training and 40% for testing.

Data Set	m-RDP	BackProp	BP 1out	CCMout
1	13.00	22.00	7.00	51.00
2	12.00	22.00	7.00	50.00
3	13.00	22.00	7.00	50.00
4	14.00	22.00	7.00	47.00
5	14.00	22.00	7.00	51.00
6	15.00	22.00	7.00	41.00
7	16.00	22.00	7.00	57.00
8	13.00	22.00	7.00	47.00
9	15.00	22.00	7.00	68.00
10	13.00	22.00	7.00	46.00
Mean	13.8	22	7	7.57
Std	1.23	0	0	7.33

Table 21 Results obtained with the m-class, and backpropagation, using the Glass data set benchmark in terms of the topology size (number of hidden/intermediate neurons) with 80% of the data used for training and 20% for testing.

Data Set	m-RDP	BackProp	BP 1out	CC Mout
1	18.00	32.00	16.00	47
2	19.00	32.00	16.00	49
3	18.00	32.00	16.00	55
4	16.00	32.00	16.00	55
5	20.00	32.00	16.00	44
6	18.00	32.00	16.00	53
7	17.00	32.00	16.00	61
8	20.00	32.00	16.00	59
9	17.00	32.00	16.00	53
10	20.00	32.00	16.00	47
Mean	18.3	32	16	52.3
Std	1.42	0	0	5.50

Table 22 Results obtained with the m-class, and backpropagation, using the Wine data set benchmark in terms of the topology size (number of hidden/intermediate neurons) with 60% of the data used for training and 40% for testing.

Data Set	m-RDP	BackProp	BP 1out	CCMout
1	4.00	28.00	31.00	6.00
2	4.00	28.00	31.00	93.00
3	4.00	28.00	31.00	50.00
4	4.00	28.00	31.00	89.00
5	4.00	28.00	31.00	123.00
6	4.00	28.00	31.00	86.00
7	4.00	28.00	31.00	99.00
8	4.00	28.00	31.00	123.00
9	4.00	28.00	31.00	10.00
10	4.00	28.00	31.00	32.00
Mean	4	28	31	71.1
Std	0	0	0	43.63

Table 23 Results obtained with the m-class, and backpropagation, using the Wine data set benchmark in terms of the topology size (number of hidden/intermediate neurons) with 80% of the data used for training and 20% for testing.

Data Set	m-RDP	BP 1out	CC Mout
1	4.00	31.00	123
2	4.00	31.00	123
3	4.00	31.00	123
4	4.00	31.00	107
5	4.00	31.00	59
6	4.00	31.00	123
7	4.00	31.00	123
8	4.00	31.00	60
9	4.00	31.00	76
10	4.00	31.00	123
Mean	4	31	104
Std	0	0	27.73

Table 24 Results obtained with the m-class, and backpropagation, using the Zoo data set benchmark in terms of the topology size (number of hidden/intermediate neurons) with 60% of the data used for training and 40% for testing.

Data Set	m-RDP	BackProp	BP 1out	CCMout	CC out
1	8.00	32.00	7.00	7.00	1.00
2	8.00	32.00	7.00	7.00	1.00
3	8.00	32.00	7.00	7.00	1.00
4	8.00	32.00	7.00	7.00	1.00
5	8.00	32.00	7.00	7.00	1.00
6	8.00	32.00	7.00	7.00	1.00
7	8.00	32.00	7.00	7.00	1.00
8	8.00	32.00	7.00	7.00	1.00
9	8.00	32.00	7.00	7.00	1.00
10	8.00	32.00	7.00	7.00	1.00
Mean	8	32	7	7	0
Std	0	0	0	0	0

Table 25 Results obtained with the m-class, and backpropagation, using the Zoo data set benchmark in terms of the topology size (number of hidden/intermediate neurons) with 80% of the data used for training and 20% for testing.

Data Set	m-RDP	BackProp	BP 1out	CC Mout	CC 1out
1	8.00	27.00	3.00	7	1
2	8.00	27.00	3.00	7	1
3	8.00	27.00	3.00	7	1
4	8.00	27.00	3.00	7	1
5	8.00	27.00	3.00	7	1
6	8.00	27.00	3.00	7	1
7	8.00	27.00	3.00	7	1
8	8.00	27.00	3.00	7	1
9	8.00	27.00	3.00	7	1
10	8.00	27.00	3.00	7	1
Mean	8	27	3	7	0
Std	0	0	0	0	0

Table 26 Results obtained with the m-class, and backpropagation, using the Iris data set benchmark in terms of the topology size (number of hidden/intermediate neurons) with 60% of the data used for training and 40% for testing.

Data Set	m-RDP	BackProp	BP 1out	Cc Mout	CC 1out
1	4.00	9.00	5.00	6.00	12.00
2	4.00	9.00	5.00	7.00	13.00
3	4.00	9.00	5.00	6.00	10.00
4	5.00	9.00	5.00	8.00	13.00
5	5.00	9.00	5.00	6.00	11.00
6	5.00	9.00	5.00	8.00	11.00
7	4.00	9.00	5.00	6.00	13.00
8	4.00	9.00	5.00	6.00	13.00
9	4.00	9.00	5.00	6.00	13.00
10	4.00	9.00	5.00	5.00	10.00
Mean	4.3	9	5	6.4	11.9
Std	0.48	0	0	0.97	

Table 27 Results obtained with the m-class, and backpropagation, using the Iris data set benchmark in terms of the topology size (number of hidden/intermediate neurons) with 80% of the data used for training and 20% for testing.

Data Set	m-RDP	BackProp	BP 1out	CC Mout	CC 1out
1	4.00	13.00	5.00	4	14
2	4.00	13.00	5.00	6	15
3	6.00	13.00	5.00	8	15
4	6.00	13.00	5.00	9	16
5	6.00	13.00	5.00	8	16
6	6.00	13.00	5.00	7	15
7	6.00	13.00	5.00	10	16
8	7.00	13.00	5.00	8	16
9	6.00	13.00	5.00	8	16
10	5.00	13.00	5.00	6	13
Mean	5.6	13	5	7.4	15.2
Std	0.97	0	0	1.71	1.03

Table 28 Results obtained with the m-class, and backpropagation, using the Soybean data set benchmark in terms of the topology size (number of hidden/intermediate neurons) with 60% of the data used for training and 40% for testing.

Data Set	m-RDP	BackProp	BP 1out	CCMout	CC 1out
1	19.00	60.00	51.00	17.00	15.00
2	17.00	60.00	51.00	17.00	15.00
3	18.00	60.00	51.00	17.00	15.00
4	16.00	60.00	51.00	17.00	14.00
5	17.00	60.00	51.00	16.00	14.00
6	16.00	60.00	51.00	16.00	14.00
7	16.00	60.00	51.00	16.00	15.00
8	16.00	60.00	51.00	16.00	14.00
9	16.00	60.00	51.00	16.00	14.00
10	16.00	60.00	51.00	17.00	15.00
Mean	16.7	60	51	16.5	14.5
Std	1.06	0	0	0.53	0.53

Table 29 Results obtained with the m-class, and backpropagation, using the Soybean data set benchmark in terms of the topology size (number of hidden/intermediate neurons) with 80% of the data used for training and 20% for testing.

Data Set	m-RDP	BackProp	BP 1out	CC Mout	CC 1out
1	17.00	50.00	51.00	17	18
2	19.00	50.00	51.00	17	17
3	18.00	50.00	51.00	18	18
4	17.00	50.00	51.00	17	18
5	17.00	50.00	51.00	17	17
6	19.00	50.00	51.00	17	18
7	17.00	50.00	51.00	17	18
8	18.00	50.00	51.00	18	17
9	17.00	50.00	51.00	18	18
10	17.00	50.00	51.00	18	19
Mean	17.6	50	51	17.4	17.8
Std	0.84	0	0	0.52	0.63

Table 30 Results obtained with the m-class, and backpropagation, using the Wisconsin Breast Cancer data set benchmark in terms of the topology size (number of hidden/intermediate neurons) with 60% of the data used for training and 40% for testing.

Data Set	m-RDP	BackProp	CC
1	10.00	16.00	6.00
2	7.00	16.00	5.00
3	9.00	16.00	5.00
4	10.00	16.00	7.00
5	11.00	16.00	8.00
6	12.00	16.00	7.00
7	11.00	16.00	7.00
8	9.00	16.00	7.00
9	10.00	16.00	6.00
10	11.00	16.00	6.00
Mean	10	16	6.4
Std	1.41	0	0.97

Table 31 Results obtained with the m-class, and backpropagation, using the Wisconsin Breast Cancer data set benchmark in terms of the topology size (number of hidden/intermediate neurons) with 80% of the data used for training and 20% for testing.

Data Set	m-RDP	BackProp	CC
1	11.00	16.00	7
2	11.00	16.00	8
3	12.00	16.00	8
4	14.00	16.00	8
5	13.00	16.00	9
6	12.00	16.00	7
7	14.00	16.00	8
8	12.00	16.00	7
9	11.00	16.00	6
10	10.00	16.00	8
Mean	12	16	7.67
Std	1.33	0	0.87

References

1. Bennett, K.P., Mangasarian, O.L.: Robust linear programming discrimination of two linearly inseparable sets. Optimization Methods and Software 1, 23–34 (1992)
2. Dasarathy, B.W.: Nosing around the neighborhood: A new system structure and classification rule for recognition in partially exposed environments. IEEE Transactions on Pattern Analysis and Machine Intelligence 2(1), 67–71 (1980)
3. Newman, D.J., Hettich, S., Merz, C.B.: C.: UCI repository of machine learning databases (1998), `http://www.ics.uci.edu/~mlearn/MLRepository.html`
4. Elizondo, D.: The recursive determinist perceptron (rdp) and topology reduction strategies for neural networks. Ph.D. thesis, Université Louis Pasteur, Strasbourg, France (1997)
5. Elizondo, D.: Searching for linearly separable subsets using the class of linear separability method. In: Proceedings of the IEEE-IJCNN, pp. 955–960 (2004)
6. Elizondo, D.: The linear separability problem: Some testing methods. Accepted for Publication: IEEE TNN 17, 330–344 (2006)
7. Fisher, R.A.: The use of multiple measurements in taxonomic problems. Annual Eugenics 7(II), 179–188 (1936)
8. Gates, G.W.: The reduced nearest neighbor rule. IEEE Transactions on Information Theory, 431–433 (1972)
9. Mangasarian, O.L., Wolberg, W.H.: Cancer diagnosis via linear programming. SIAM News 23(5), 1–18 (1990)
10. Preparata, F.P., Shamos, M.I.: Computational Geometry. An Introduction. Springer, New York (1985)
11. Rosenblatt, F.: Principles of Neurodynamics. Spartan, Washington D.C (1962)
12. Tajine, M., Elizondo, D.: Enhancing the perceptron neural network by using functional composition. Tech. Rep. 96-07, Computer Science Department, Université Louis Pasteur, Strasbourg, France (1996)
13. Tajine, M., Elizondo, D., Fiesler, E., Korczak, J.: Adapting the $2-$class recursive deterministic perceptron neural network to $m-$classes. In: The International Conference on Neural Networks (ICNN), IEEE, Los Alamitos (1997)
14. Weiss, S.M., Kulikowski, C.A.: Computer Systems That Learn. Morgan Kaufmann Publishers, San Mateo (1991)
15. Wolberg, W.H., Mangasarian, O.: Multisurface method of pattern separation for medical diagnosis applied to breast cytology. Proceedings of the National Academy of Sciences 87, 9193–9196 (1990)