Negative Correlation in Incremental Learning

L. Minku, H. Inoue[*]and X. Yao

The Centre of Excellence for Research in
Computational Intelligence and Applications (CERCIA),
School of Computer Science, The University of Birmingham,
Edgbaston, Birmingham B15 2TT, UK
Phone: +44 121 414 3747, Fax: +44 121 414 2799
{F.L.Minku,H.Inoue,X.Yao}@cs.bham.ac.uk

26 August 2007

[*]Hirotaka Inoue is also with the Department of Electrical Engineering and Information Science, Kure National College of Technology, 2-2-11 Agaminami, Kure-shi, Hiroshima 737-8506, Japan.

Abstract

Negative Correlation Learning (NCL) has been successfully applied to construct neural network ensembles. It encourages the neural networks that compose the ensemble to be different from each other and, at the same time, accurate. The difference among the neural networks that compose an ensemble is a desirable feature to perform incremental learning, for some of the neural networks can be able to adapt faster and better to new data than the others. So, NCL is a potentially powerful approach to incremental learning. With this in mind, this paper presents an analysis of NCL, aiming at determining its weak and strong points to incremental learning. The analysis shows that it is possible to use NCL to overcome catastrophic forgetting, an important problem related to incremental learning. However, when catastrophic forgetting is very low, no advantage of using more than one neural network of the ensemble to learn new data is taken and the test error is high. When all the neural networks are used to learn new data, some of them can indeed adapt better than the others, but a higher catastrophic forgetting is obtained. In this way, it is important to find a trade-off between overcoming catastrophic forgetting and using an entire ensemble to learn new data. The NCL results are comparable with other approaches which were specifically designed to incremental learning. Thus, the study presented in this work reveals encouraging results with negative correlation in incremental learning, showing that NCL is a promising approach to incremental learning.

Keywords

Neural network ensembles, incremental learning, negative correlation learning, multi-layer perceptrons, self-generating neural tree, self-organizing neural grove, classification.

List of Abbreviations

NCL	Negative Correlation Learning
SGNT	Self-Generating Neural Tree
SGNN	Self-Generating Neural Network
ESGNN	Ensemble of Self-Generating Neural Networks
SONG	Self-Organizing Neural Grove
MLP	Multi-Layer Perceptron
SOM	Self-Organizing Map
EFuNN	Evolving Fuzzy Neural Network
AdaBoost	Adaptive Boosting
ART	Adaptive Resonance Theory
GL	Generalization Loss

1 Introduction

Neural network ensembles have been showing to improve single neural networks accuracy. In the last years many ensembles methods were developed to construct neural network ensembles, e.g., Boosting (Schapire; 1990; Freund and Schapire; 1997) and Bagging (Breiman; 1996). However, in most of the methods developed until now, the training procedure needs all training data simultaneously to perform learning and no posterior learning is possible. This restriction can be a problem to update the ensembles when new data is available after a initial training or when the data sets are large and cannot be loaded in memory at one go.

An incremental learning algorithm gives a system the ability to learn from new information as it becomes available (Seipone and Bullinaria; 2005). It has to meet the following criteria (Polikar et al.; 2001):

1. It should be able to learn additional information from new data.
2. It should not require access to the original data, used to train the existing classifier.
3. It should preserve previously acquired knowledge (that is, it should not suffer from catastrophic forgetting).
4. It should be able to accommodate new classes that may be introduced with new data.

There are several algorithms proposed to perform incremental learning, although many of them do not satisfy all the criteria to be considered incremental learning approaches. Two approaches that are notable for satisfying all the necessary criteria are (fuzzy) Adaptive Resonance Theory modules Map (ARTMAP) (Carpenter et al.; 1991, 1992) and Evolving Fuzzy Neural Networks (EFuNNs) (Kasabov; 2001). Other two important approaches to incremental learning are Learn++ (Polikar et al.; 2001) and Evolved Incremental Learning for Neural Networks (Seipone and Bullinaria; 2005). The former satisfies all the necessary criteria to incremental learning and it is based on Adaptive Boosting (AdaBoost) (Schapire; 1990; Freund and Schapire; 1997), which is an ensemble method. The later is based on Evolutionary Algorithms (Eiben and Smith; 2003) and, although the neural networks produced by the evolutionary approach satisfy all the necessary criteria to incremental learning, we shall discuss in section 3 that the evolutionary algorithm used to produce them could have some problems related to the second criterion, making the approach as a whole not satisfy all the criteria. Another ensemble method which showed to be successful to incremental learning is Self-Organizing Neural Grove (SONG) (Inoue and Narihisa; 2005). These methods are described in section 3.

Neural network ensembles are potentially important methods to perform incremental learning. Research about neural network ensembles to incremental learning is very recent and constitute a field that is not extensively explored yet. Ensembles are potentially important methods to incremental learning because, as they are constituted by several different neural networks, some of them can be able to adapt faster and better to new data than the others. In this way, the neural networks which adapt better can make the ensemble overcome the problem of the neural networks which could not have a good adaptation to new data. It is important to observe that, for different incoming data, different neural networks can be the ones which will have the best adaptation. In this way, the neural network which had the best adaptation to a certain data set may not be the neural network which will have the best adaptation to another incoming data set. Further discussion related to this topic is presented in sections 4.2.4 and 4.2.5.

Nevertheless, it is necessary that the ensemble is diverse for that the differences among its neural networks make them adapt in different ways to new data. Negative Correlation Learning (NCL) (Liu and Yao; 1999a,b) is a successful approach to construct neural network ensembles. It has formally and empirically shown to encourage the neural networks that compose the ensemble to be different from each other and, at the same time, accurate. So, NCL is a potentially powerful approach to incremental learning.

In off-line mode, ensemble learning approaches which directly encourage diversity, such as NCL, have been showing to outperform other ensemble methods (Islam et al.; 2003; Wang et al.; 2004; Chandra and Yao; 2006). So, it is important to explore the advantages of NCL in an incremental setting too. It is woth to notice that the current incremental ensemble learning approaches do not directly encourage diversity.

With this in mind, this paper investigates the use of negative correlation in incremental learning, in order to determine its strong and weak points to incremental learning. Negative correlation was never used to incremental learning before, so it is very important to study its advantages and disadvantages to incremental learning. Two different approaches to use negative correlation in incremental learning (called Fixed Size NCL and Growing NCL) are presented and analysed. To support the analysis, NCL is compared with SONG (Inoue and Narihisa; 2005), which is a successfully approach to incremental learning.

The analysis shows that it is possible to use negative correlation in incremental learning, although each approach also has its weakness. It shows that some of the networks of the Fixed Size NCL ensemble can indeed adapt better than the others, and they are not the same networks for different incoming data. However, this approach suffers more catastrophic forgetting than Growing NCL. The analysis also shows that it is possible to use Growing NCL to overcome catastrophic forgetting, an important problem related to incremental learning. However, this approach has low generalization in comparison with Fixed Size NCL and do not take advantage of using more than one neural network of the ensemble to learn new data. In this way, it is important to find a trade-off between overcoming catastrophic forgetting and using an entire ensemble to learn new data. The improvement in the generalization after training with new data for both the approaches is comparable with other approaches specifically developed to incremental learning. The study reveals encouraging results with negative correlation in incremental learning and shows that NCL is a promising approach to incremental learning.

The rest of the paper is organized as follows. Section 2 describes NCL and how it can be used to incremental learning. Section 3 presents some related works on incremental learning. Section 4 presents the experimental study of negative correlation in incremental learning. Section 5 presents some discussion about the relationship between diversity and generalization. Section 6 presents the conclusions and future works.

2 Negative Correlation Learning

This section explains the basic ideas of Negative Correlation Learning (NCL) (Liu and Yao; 1999a,b). For more details about the theoretical basis of NCL, it is recommendable to read (Chandra et al.; 2006) and (Brown; 2004).

Given a training set T of size N:

$$T = \{(x(1), d(1)), (x(2), d(2)), ..., (x(N), d(N))\} \ ,$$

where x ($x \in \mathbb{R}^p$) is the input to a neural network, d is the desired output and is a scalar. The assumption that d is a scalar is made to simplify the exposition of the ideas without loss of generality. Consider estimating d by forming an ensemble whose output is a simple average of a set of M neural network outputs[1]:

$$F(n) = \frac{1}{M} \sum_{i=1}^{M} F_i(n) \ , \tag{1}$$

where $F_i(n)$ and $F(n)$ are the output of the ith individual neural network and the ensemble, respectively, on the nth training pattern.

The aim of NCL is to produce diverse neural networks in an ensemble, by inserting a penalty term into the error function of each individual neural network in the ensemble. All individual neural networks in the ensemble are trained simultaneously and interactively on the same

[1] During negative correlation learning, simple average is used to combine the neural network outputs. However, the combination method used by the ensemble during the test phase can be another one, e.g., majority vote.

training data set T. The error function E_i for the ith neural network in NCL is defined by the following equation:

$$E_i = \frac{1}{N}\sum_{n=1}^{N} E_i(n) = \frac{1}{N}\sum_{n=1}^{N}\left(\frac{1}{2}(F_i(n) - d(n))^2 + \gamma p_i(n)\right) \ , \tag{2}$$

where $E_i(n)$ is the error of the ith neural network after the presentation of the nth training pattern. The first term in equation 2 is the empirical risk function of the ith neural network. The second term p_i is the correlation penalty function. The purpose of minimizing p_i is to penalize positive correlation of errors from different neural networks, i.e. , to encourage negative correlation of a neural network error with the error of the rest of the ensemble. The parameter γ is used to adjust the strength of the penalty and it is problem-dependent (Brown et al.; 2005). The penalty function p_i may use the following equation:

$$p_i(n) = (F_i(n) - F(n))\sum_{i \neq j}(F_j(n) - F(n)) \ . \tag{3}$$

The partial derivative of $E_i(n)$ with respect to the output of the network i on the nth training pattern is:

$$\frac{\partial E_i(n)}{\partial F_i(n)} = F_i(n) - d(n) - \gamma\left[2\left(1 - \frac{1}{M}\right)(F_i(n) - F(n))\right] \ . \tag{4}$$

When M is large, $(1-1/M)$ equals to 1. The standard Back-propagation algorithm (Rumelhart et al.; 1986) can be used with equation 4 for weight adjustments of the neural networks, which can be Multi-Layer Perceptrons (MLPs). The weight updates of all neural networks is performed simultaneously.

In previous papers, as (Liu and Yao; 1999a), the partial derivative was calculated as:

$$\frac{\partial E_i(n)}{\partial F_i(n)} = F_i(n) - d(n) - \lambda(F_i(n) - F(n)) \ . \tag{5}$$

Nevertheless, this calculation considers that $F(n)$ is constant with respect to $F_i(n)$, when actually it is not. Considering equation 5, it was believed that the strength parameter, which was called λ, was entirely problem-dependent. The strength parameter γ was introduced by Brown et al. (2005) to calculate the partial derivative correctly. Its relation to λ is shown in equation 6. As it can be seen, λ is not entirely problem-dependent. It has a deterministic component: $2(1 - \frac{1}{M})$. The parameter γ is still problem-dependent.

$$\lambda = \gamma \left[2 \left(1 - \frac{1}{M} \right) \right] \ . \tag{6}$$

Brown et al. (2005) mathematically showed that γ has an upper bound (equation 7). The upper bound is used to avoid loosing useful gradient information when negative correlation is used. The negative correlation penalty term 'warps' the error landscape of the neural network, making the global optimum hopefully easier to locate. However, if the landscape is warped too much, it could eliminate any useful gradient information.

$$\gamma_{upper} = \frac{M^2}{2(M-1)^2} \ . \tag{7}$$

The following observations can be made from equations 2, 3 and 4:

- During the training process, all individual neural networks interact with each other through their penalty terms in the error functions. Each neural network minimizes not only the difference between $F_i(n)$ and $d(n)$, but also the difference between $F(n)$ and $d(n)$, considering the error of all other neural networks while training a particular neural network.
- For $\gamma = 0$, the individual neural networks are trained independently.
- For $\gamma = 1$, we get from equation 4:

  $$\frac{\partial E_i(n)}{\partial F_i(n)} = \left(2 - \frac{2}{M} \right) F(n) + \left(-1 + \frac{2}{M} \right) F_i(n) - d(n) \ . \tag{8}$$

  The training of the neural networks minimizes the difference between $F(n)$ and $d(n)$ as well as the difference between $F_i(n)$ and $d(n)$. However, the minimization of $Fi(n) - d(n)$ is the cause for the minimization of $F(n) - d(n)$, as $F(n)$ is obtained from equation 1. So, it is possible to consider that $F_i(n)$ tends to $d(n)$. In this way, from equation 8, we get:

  $$\frac{\partial E_i(n)}{\partial F_i(n)} = \left(2 - \frac{2}{M} \right) F(n) + \left(-2 + \frac{2}{M} \right) d(n) = \left(2 - \frac{2}{M} \right) (F(n) - d(n)) \ . \tag{9}$$

  When M is large, we get:

  $$\frac{\partial E_i(n)}{\partial F_i(n)} = 2(F(n) - d(n)) \ . \tag{10}$$

  Note that the empirical risk function of the ensemble for the nth training pattern is:

$$E_{ens}(n) = \frac{1}{2}\left(\frac{1}{M}\sum_{i=1}^{M} F_i(n) - d(n)\right)^2 \quad , \tag{11}$$

where M indicates the number of neural networks in the ensemble.

The partial derivative of $E_{ens}(n)$ with respect to $F_i(n)$ on the nth training pattern is:

$$\frac{\partial E_{ens}(n)}{\partial F_i(n)} = \frac{1}{M}\left(\frac{1}{M}\sum_{j=1}^{M} F_j(n) - d(n)\right) = \frac{1}{M}(F(n) - d(n)) \ . \tag{12}$$

In this case, we get:

$$\frac{\partial E_i(n)}{\partial F_i(n)} = (2M-2)\frac{\partial E_{ens}(n)}{\partial F_i(n)} \ . \tag{13}$$

And, when M is large:

$$\frac{\partial E_i(n)}{\partial F_i(n)} = 2M\frac{\partial E_{ens}(n)}{\partial F_i(n)} \ . \tag{14}$$

In other words, the minimization of the empirical risk function of the ensemble can be achieved by minimizing the error functions of individual neural networks. In effect, a large and more complex task of training the ensemble is automatically decomposed into a number of simpler tasks of training individual neural networks.

2.1 *Using Negative Correlation in Incremental Learning*

Two approaches to utilize negative correlation in incremental learning are introduced and analysed in this paper. One of them is called Fixed Size NCL and the other one is called Growing (Size) NCL. These approaches are explained in sections 2.1.1 and 2.1.2.

2.1.1 Fixed Size NCL

A direct way of performing incremental learning using NCL is to create an ensemble of neural networks and train it with the first available data set. When another data set comes, the initial weights of the networks that compose the ensemble are the weights obtained with the previous training and the whole ensemble is trained with the new data set. No training is performed with old data.

This approach has the advantage that it allows all the networks train with the incoming data. In this way, some of the neural networks can adapt fast and better than others and it is expected that these neural networks make the ensemble overcome the problem of the neural networks which are not able to have a good adaptation to the new data. Besides, as the entire ensemble is always used to all incoming data, if there are similarities among new data and old data, the training error in the first epoch of training with the new data can be lower. However, it is possible that the neural networks suffer from catastrophic forgetting. The advantages and disadvantages of Fixed Size NCL are further discussed in section 4.

2.1.2 Growing NCL

Another way to perform incremental learning using NCL is to create an ensemble that has initially only one neural network. The neural network is trained with the first available data set. To each new incoming data set, a new neural network is inserted in the ensemble. Only the new neural network is trained with the new data set. The other neural networks that were previously inserted in the ensemble do not receive any new training on the new incoming data, but their outputs to the new data are calculated in order to interact with the new MLP, which is trained using NCL.

This approach has the advantage that it can tackle the problem of catastrophic forgetting, as each neural network is trained with only one incoming data set. As it trains only one neural network, instead of the whole ensemble, with each new data set, its learning is faster than Fixed Size NCL. However, only one neural network to each of the data sets could be a short number to attain a good accuracy. The advantages and disadvantages of Growing NCL are further discussed in section 4.

3 Related Work on Incremental Learning

This section briefly describes some incremental algorithms and their differences with negative correlation in incremental learning. The described approaches are (fuzzy) ARTMAP, EFuNNs, Learn++ and Evolved Incremental Learning for Neural Networks.

Carpenter et al. (1991) proposed a neural network architecture called ARTMAP. It is constructed from a pair of Adaptive Resonance Theory (ART) modules (ART_a and ART_b) which are capable of self-organizing categories in response to sequences of input patterns. ART_a receives the inputs of the training patterns and ART_b receives the targets. These modules are linked by an associative learning network and the system can continue learning one or more data sets, without degrading previous learned data sets. Another neural network architecture was also proposed by Carpenter et al. (1992), combining fuzzy logic and ART. Both the approaches are based on the generation of new clusters in response to new patterns that are sufficiently different from previously seen patterns. According to Polikar et al. (2001):

> Each cluster learns a different hyper-rectangle shaped portion of the feature space in an unsupervised mode, which are then mapped to target classes. Since previously generated clusters are always retained, ARTMAP does not suffer from catastrophic forgetting. Furthermore, ARTMAP does not require access to previously seen data, and it can accommodate new classes.

Some problems of ARTMAP are the high sensibility to the parameter which controls the sufficiency of difference of new patterns to previously seen patterns, the possibility to overfit training data if this parameter is not properly chosen and the sensibility to noises and order of presentation of the training data.

Kasabov (2001) proposed another incremental learning neural network architecture, called Evolving Fuzzy Neural Network (EFuNN). The term *evolving* is related to the fact that its structure continuously adapt to the environment, and not to the use of evolutionary algorithms, as it is usually used in the literature. As fuzzy ARTMAP, the architecture also join the expressive power of fuzzy logic to the neural networks functional characteristics. The learning occurs in an on-line, incremental, fast (one-pass through the data) and local mode. It learn associations between clusters of the fuzzy input space of the problem and clusters of the fuzzy output space. The clusters are represented by hyperspheres and new clusters are created when a new pattern is sufficiently different from previously seen patterns. EFuNN also does not require access to previously seen data and can accommodate new classes.

The parameter which indicate whether a new pattern is too different from the existent output clusters (which is the radius of the output hyperspheres) and the initial radius of the input hyperspheres are the parameters which have highest influence in the EFuNN training (Zanchettin et al.; 2005). In the same way as with ARTMAP, it is possible to overfit training data if these parameters are not properly chosen.

We can observe that, instead of creating new clusters to incoming data (as ARTMAP and EFuNN), Growing NCL creates a new MLP to join the ensemble every time that a new training set is available. However, each new MLP is created to be trained with the whole new training set and not only with examples that are sufficiently different from previously presented examples. In this way, it is possible to avoid the problem of tunning parameters related to the sufficiency of difference and the problem to find a measure of difference among old and new patterns to be used with MLPs. Besides, MLPs previously inserted in the ensemble do not receive any training with new data in order to avoid catastrophic forgetting, which is a problem of MLPs to incremental learning. So, the Growing NCL can overcome the MLP problem of catastrophic forgetting.

Another notable approach to incremental learning is Learn++ (Polikar et al.; 2001). It is inspired on AdaBoost (Schapire; 1990; Freund and Schapire; 1997). Instead of creating new clusters for previously unseen data portions of the feature space (as ARTMAP and EFuNN), Learn++ create multiple classifiers. As in AdaBoost, patterns of the data set are sampled according to a distribution of probability in order to create the training set to train the classifiers in a sequential way. However, in AdaBoost, the distribution of probability is built in a way to give higher priority to instances misclassified only by the last previously created classifier (the errors of the classifiers created before the last one are indirectly considered, although also each one in an isolated way). In Learn++, the distribution is created considering the misclassification by the **composite** hypothesis, formed by all the classifiers created so far to that incoming data set. In this way, incremental learning is possible, particularly when instances from new classes are introduced. Learn++ also does not require access to previously seen data.

An important difference between Learn++ and Fixed Size NCL is that NCL directly encourages diversity among the neural networks of the ensemble, by using the penalty term in the error function of their learning algorithms. This is an important characteristic to incremental learning, as it is this difference that will allow the neural networks to adapt differently to new incoming data and some better than the others, depending on the data. Learn++ does not use this valuable characteristic of ensembles to incremental learning, as it does not use multiple classifiers trained with old data to learn new data. It always creates new multiple classifiers to that.

An important difference between Learn++ and Growing NCL is that, although Learn++ creates each of the classifiers to the new data set using a probability of distribution that is related to all the previous classifiers created to the new data set, the construction of classifiers to the new data set does not have interaction with the classifiers created to previous data sets. In NCL, the new neural networks that are inserted in the ensemble are trained interacting with the old neural networks, which were trained with the previous data sets. So, the new neural networks are encouraged to be different from the previously learned neural networks. In this way, if there are similarities among new and old data, the new neural network can contribute to the ensemble as a whole by giving outputs that may be different from the outputs of the previously learned neural networks. It is important to bear in mind that an successfully ensemble must have neural networks with error rates below 0.5 and whose errors are at least somewhat not correlated (Dietterich; 1997). Training the new neural network using NCL penalizes the correlation of the new neural network with the previous trained neural networks. Another difference between Learn++ and Growing NCL is that Learn++ creates multiple classifiers when new data sets are presented, while Growing NCL creates only one MLP to each new data set, reducing the necessary computational effort to perform training. However, it may be possible that inserting only one MLP to the ensemble will not lead to a accuracy as good as if multiple MLPs were inserted. This issue is discussed in section 4.

Recently, Seipone and Bullinaria (2005) developed a new approach to incremental learning using evolving neural networks. One of the advantages of evolving neural networks is that it is possible to tune any of the parameters of the neural network using the evolutionary algorithm, solving the hard problem of determining the parameters by hand. Besides, evolving neural networks can adapt to dynamic environments, changing their architecture and learning rules appropriately. In Seipone and Bullinaria (2005)'s work, the authors use an evolutionary algorithm to evolve some MLP parameters, as the learning rates, initial weight distributions and error tolerance. The evolutionary process aims at evolving the parameters to produce networks with better incremental abilities. In each generation, the neural networks with the parameters codified by the evolutionary algorithms are trained using first the training patterns of a particular data set. After that, the neural networks are trained with another data set and so on. While a neural network is being trained with a particular data set, the other data sets are not used in the training. The error produced by testing the neural networks with a validation set is used as a fitness measure. The approach is also able to cope with new classes of data.

The evolutionary approach to incremental learning introduced in (Seipone and Bullinaria; 2005) has the problem that, although each neural network is trained with only one data set at time, the evolutionary algorithm considers all the data sets in each generation of the evolution. So, the networks satisfy all the necessary criteria to incremental learning, but the evolutionary algorithm does not. This could be considered a problem related to the second necessary condition to incremental learning, presented in section 1.

Self-Organizing Neural Grove (SONG) (Inoue and Narihisa; 2003) is an ensemble of Self-Generating Neural Networks (SGNNs) (Wen et al.; 1992) which uses a pruning method for the structure of the SGNNs to reduce the computation time and the memory capacity of Ensembles of Self-Generating Neural Networks (ESGNNs) (Inoue and Narihisa; 2000). SGNNs have a simple network design and high speed learning. They are an extension of the self-organizing maps (SOM) of Kohonen (Kohonen; 1995) and utilize competitive learning, implemented as a Self-Generating Neural Tree (SGNT).

SGNTs can be constructed directly from the given training data, without any intervening human effort. The SGNT algorithm is defined as the problem of how to construct a tree structure from the given data which consist of multiple attributes under the condition that the final leaves correspond to the given data. An ESGNN is constructed by presenting different orders of the training set to each of the SGNTs. The output of the ensemble is the majority vote of the outputs of the SGNTs. After the construction of the ensemble, it is possible to apply a pruning method to compose a SONG. For details about the SGNT algorithm and the pruning method, it is recommendable to read (Inoue and Narihisa; 2005).

SONG can be directly applied to incremental learning and it has shown to be a successful approach to do that (Inoue and Narihisa; 2005). It joins the advantages of neural network ensembles with the advantage of using a base classifier that is proper to incremental learning.

Similarly to Fixed Size NCL, SONG also train all its components with all the data sets which are presented to the ensemble. So, it also takes advantage of the differences among the components of the ensemble, which make some of the components adapt better than the others, depending on the incoming data. However, the SONG learning algorithm does not encourage diversity in the ensemble, as NCL. The differences among the SGNTs are only produced by the presentation of the data set in different orders. In NCL, besides the presentation of the data set in different orders to each MLP, the diversity among the neural networks is encouraged by the penalty term, as explained in section 2 and formally shown in (Chandra et al.; 2006). The number of SGNTs in SONG is fixed and does not increase, as in Fixed Size NCL.

Another interesting difference between SONG and NCL is that SONG uses a base classifier that is proper to incremental learning, while NCL uses MLPs, which (when used as single classifiers) do not have a good behaviour to incremental learning. In this way, it is interesting to analyse whether Growing NCL is able to overcome the problem of catastrophic forgetting better than SONG and whether the Fixed Size NCL suffer more from catastrophic forgetting than SONG.

SONG has a fast training in comparison with NCL, for the SGNTs use one-pass learning, while the MLPs need the presentation of the training data for a certain number of epochs. However, the size of the MLPs is fixed and pre-determined, while the size of the SGNTs can increase as new patterns are presented. So, it is possible that the SGNTs of SONG have a higher size than the MLPs of NCL, increasing its testing time.

As SONG joins the advantages of a base classifier which is proper to incremental learning to the advantages of using an ensemble to perform incremental learning and it has shown to be a successful incremental approach, in order to support the negative correlation in incremental learning analysis, it was chosen as the main approach to be compared with NCL.

4 Experimental Study of Negative Correlation in Incremental Learning

This section describes the experiments which were made with the NCL approaches to incremental learning. An extensive analysis, considering various aspects of the results, is made aiming at determining and discussing the strong and weak points of negative correlation in incremental learning. To support the analysis, NCL is compared with SONG, which is a successfully approach to incremental learning that joins the advantages of a base classifier which is proper to incremental learning to the advantages of using an ensemble to perform incremental learning. Negative correlation in incremental learning was also compared with single MLPs, Learn++ and the evolutionary approach introduced by Seipone and Bullinaria (2005).

The analysis shows that it is possible to use negative correlation in incremental learning, although each approach also has its weakness. It shows that some of the networks of the Fixed Size NCL ensemble can indeed adapt better than the others, and they are not the same networks for different incoming data. However, this approach suffers more catastrophic forgetting than Growing NCL. The analysis also shows that it is possible to use Growing NCL to overcome catastrophic forgetting, an important problem related to incremental learning. However, this approach has low generalization in comparison with Fixed Size NCL and do not take advantage of using more than one neural network of the ensemble to learn new data. The improvement in the generalization after training with new data for both the approaches is comparable with other approaches specifically developed to incremental learning. The study reveals encouraging results with negative correlation in incremental learning and shows that NCL is a promising approach to incremental learning.

The experiments utilized five benchmark classification databases from the UCI Machine Learning Repository (Newman et al.; 1998): Letter, Vehicle, Optical Digits, Adult and Mushroom. Table 1 presents a summary of the databases. It shows the number of input and output attributes, the number of patterns of the database, the size (and number) of the training sets used in the incremental learning and the size of the test data sets. The signal ∼ indicates that some of the training sets contain one pattern less than the number indicated. For NCL, 1/3 of each training set was used as validation data set, to perform early stop according to the generalization loss criterion (Prechelt; 1994).

Table 1: Databases

Database	Inputs	Outputs	Patterns	Training	Test
Letter	16	26	20000	2000 (9)	2000
Vehicle	18	4	846	210 (3)	216
Optdigits	64	10	5620	200 (6)	4420
Adult	14	2	45222	∼ 4523 (9)	4523
Mushroom	21	2	8124	∼ 813 (9)	813

The Vehicle data set is considered as one of the most difficult databases in the repository, since generalization performances using various algorithms have been in the 65%-80% range (Polikar et al.; 2001). MLPs usually do not attain good test error rates to Letter data set, while methods like K-Nearest Neighbors (Larose; 2004) manage to attain good generalization (Adamczak et al.; 1997). Letter, Mushroom and Adult are databases which have more than 8000 patterns. In this way, it was possible to divide the database in various different training data sets of similar size to perform incremental learning. The sizes of the training and test sets for Vehicle and Optical Digits were chosen in order to allow comparisons with Learn++ and the evolutionary approach.

The rest of this section is divided as follows. Section 4.1 presents the parameters used in the experiments and the executions. Section 4.2 presents an extensive analysis of the results of the experiments, showing the weak and strong points of negative correlation in incremental learning.

4.1 *Experimental Setup*

The experiments are composed by 30 executions of each one of the approaches presented in table 2, to each database.

Table 2: Approaches Used in the Experiments

Single MLP	h hidden nodes
	$3h$ hidden nodes
	$5h$ hidden nodes
	$10h$ hidden nodes
Fixed Size NCL	5 MLPs with h hidden nodes each
	10 MLPs with h hidden nodes each
Growing NCL	MLPs contain h hidden nodes each
SONG	1 SGNT
	5 SGNTs
	10 SGNTs

The criteria to stop the Back-propagation learning were:

- Early stop based on the Generalization Loss (GL) (Prechelt; 1994). According to this criterion, if the generalization loss (based on the validation error) is higher then a predefined parameter α, the training stops. This criterion was considered only after a certain training progress (defined in (Prechelt; 1994)) was attained.
- Maximum number of epochs.

Table 3 summarizes some of the parameters used in the executions performed with NCL and single MLPs. It shows the number h of hidden nodes, the maximum number of epochs, the value α to the GL criterion and the minimum progress. The MLPs were trained with learning rate of 0.1, except for mushroom database, in which the learning rate was 0.05. The interval of initial values for the weights was [-1,1] and the strip size, used by the GL and training progress criteria, was 5, except for Vehicle, in which it was 10. The strength parameter γ for the NCL penalty term was 0.390625. All the parameters were empirically chosen, based on preliminary executions. The single MLPs were executed with both h, $3h$, $5h$ and $10h$ hidden nodes, although the use of $10h$ hidden nodes provides a fairer comparison with ensembles.

Table 3: NCL Parameters

Database	Hidden Nodes	Max. Epochs	GL_α	Min. Progress
Letter	40	300	1.5	10
Vehicle	30	2000	5	50
Optdigits	10	100	100	0
Adult	5	100	5	20
Mushroom	25	0.05	50	0

The executions with SONG used the parameter $\alpha = 1.0$. For more details about SONG, it is recommendable to read (Inoue and Narihisa; 2005).

4.2 *Experimental Results and Analysis*

This section presents the analyses of the results of the experiments explained in section 4.1. The analyses include an analysis of the generalization obtained by the approaches, an analysis of the improvement in the generalization from the presentation of the first to the last training sets utilized in the incremental learning, an analysis of the degradation of the accuracy on old training sets as new data sets are used for training, an analysis of the capacity that some neural networks have to adapt better than the others for new data and an analysis of the intersection of the correct response sets of each neural network of the ensemble. A comparison between NCL and other approaches than SONG and single MLPs is also presented.

4.2.1 Generalization

The generalization (1 - test classification error) of all the approaches in the test set was measured after the presentation and learning of each one of the training sets. We will call the presentation and learning of each data set as an incremental step. T-Student statistic tests (Witten and Frank; 2000) with level of significance of 5% and 1% were used to check whether there is statistically significant difference between the generalization average of each of the approaches in each incremental step.

Usually, in the literature, level of significance of 1% is used for critical applications, while 5% is used for non-critical applications. However, Dietterich (1998) concluded that T-Student tests with 5% of significance have high probability of incorrectly detecting that there is difference when no difference exists, not being recommended. So, we decided to perform tests with level of significance of 1% to reduce the probability of this problem and will consider the results of the tests which adopt 1% as more plausible.

Consider that we are comparing the approaches A and B. In this section, it is considered that the approach A is better than the approach B if the approach A contains more incremental steps with generalization averages higher than B, than B has in relation to A.

The statistical tests show that Fixed Size NCL with 5 MLPs is always either equal or worse than Fixed Size NCL with 10 MLPs when both 5% and 1% of level of significance are adopted. SONG with 5 SGNTs is considered always worse than SONG with 10 SGNTs when 5% is adopted and either equal or worse when 1% is adopted.

The comparison between single MLPs with h, $3h$ and $5h$ hidden nodes and Fixed Size NCL with 10 MLPs or Growing NCL could be considered unfair for the single MLPs or for the NCL approaches, as the total number of hidden nodes used by the single MLPs is lower than the number of nodes utilized by the other approaches[2]. With level of significance of 5%, the comparison shows that MLP with h and $3h$ hidden nodes are always worse than Fixed Size NCL with 10 MLPs. Single MLPs with $5h$ hidden nodes are equal to Fixed Size NCL with 10 MLPs in Mushroom, better in Adult, and worse in Vehicle, Letter and Optical Digits.

Considering level of significance of 1%, some of the above detected differences between the approaches are considered equalities, as expected, reducing the probability of detecting differences that do not exist. Single MLPs with h, $3h$ and $5h$ hidden nodes are either equal or worse than Fixed Size NCL with 10 MLPs.

With level of significance of both 5% and 1%, single MLPs with h, $3h$ and $5h$ hidden nodes are either equal or better than the Growing NCL. SONG with 1 SGNT is always worse than SONG with 10 SGNTs.

As the comparison between single MLPs with h, $3h$ and $5h$ hidden nodes and Fixed Size NCL with 10 MLPs or Growing NCL could be considered unfair, from this point of the paper, the analyses will always be related to the following approaches:

- Single MLP with $10h$ hidden nodes;
- Fixed Size NCL with 10 MLPs, with h hidden nodes each;
- Growing NCL in which the MLPs contain h hidden nodes each;
- SONG with 1 SGNTs.
- SONG with 10 SGNTs.

[2] It is possible that a lower number of nodes either benefit or prejudice the result of the learning, depending on the database.

The number of hidden nodes of each MLP and the number of MLPs of Fixed Size NCL will be considered implicit.

Figure 1 shows the generalization averages among the 30 executions performed with each approach, for each data set. It can be observed that, for Vehicle and Adult, the generalization averages of MLP and the NCL incremental approaches are higher (better) than the generalization averages of SONG (both 1 and 10 SGNTs). For the other databases, the generalization averages of SONG were higher. T-Student statistic tests of the generalization average in each incremental step with both 5% and 1% of level of confidence, as explained above, confirm this analysis. This result shows that the generalizations obtained by NCL are comparable (some are better and some are worse) to the generalizations obtained by SONG, when incremental learning is performed. This is important to show the applicability of negative correlation in incremental learning.

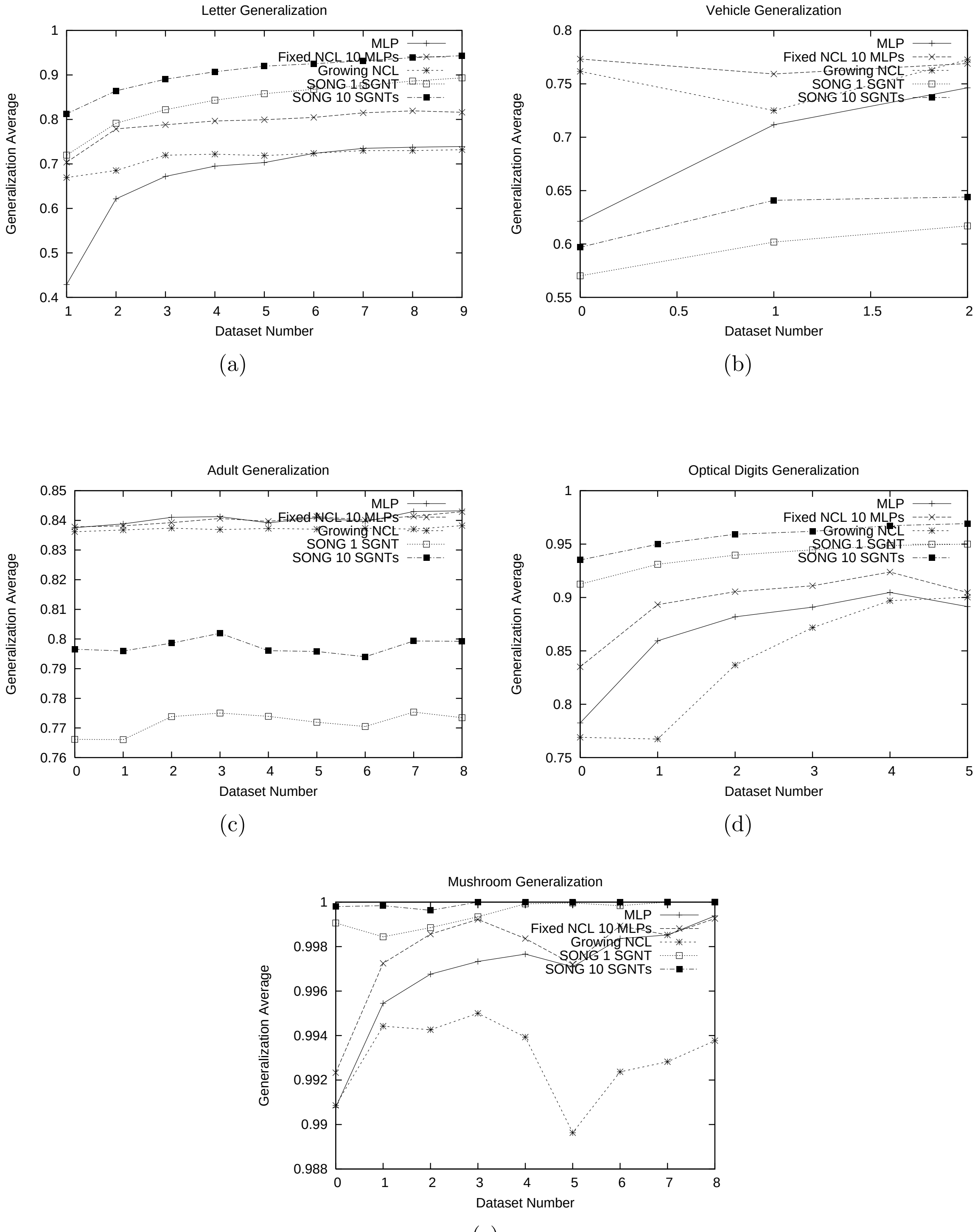
It is also important to compare the results obtained by single MLP, Fixed Size NCL and Growing NCL. Table 4 shows the rank of difficulty of the databases, from the most difficult to the easiest, considering the generalizations of the last incremental step obtained by single MLP. The table also shows the generalization averages of each approach in the first and last incremental steps, calculated among the 30 executions performed with the approaches for each database.

Table 4: Databases Difficulty - From the Most Difficult to the Easiest

Rank	Database	Generalization Av. (first-last incremental step)		
		MLP	Fixed Size NCL	Growing NCL
1	Letter	0.42900-0.73920	0.70362-0.81617	0.66928-0.73193
2	Vehicle	0.62130-0.74630	0.77330-0.76898	0.76173-0.77269
3	Adult	0.83754-0.84321	0.83781-0.84294	0.83618-0.83823
4	Optical Digits	0.78250-0.89153	0.83509-0.90464	0.76900-0.90027
5	Mushroom	0.99077-0.99938	0.99233-0.99926	0.99086-0.99377

Considering table 4 and the generalization averages (figure 1), it is possible to make the following comments about the generalization of single MLP, Fixed Size NCL and Growing NCL:

- In the most difficult databases (Letter and Vehicle), the best approach was Fixed Size NCL and the worst was MLP.
- In the easiest databases (Optical Digits and Mushroom), the best approach was also Fixed Size NCL. However, the worst was Growing NCL.
- In the Adult database, the best approach was MLP and the worst was Growing NCL. In many of the incremental steps, however, MLP had generalization considered statistically equal to Fixed Size NCL.



(a)

(b)

(c)

(d)

(e)

Figure 1: Generalization Averages

All these comments are confirmed by T-Student statistical tests with both 5% and 1% of level of confidence, as it was explained in the beginning of this section. We can observe that the best approach is usually Fixed Size NCL and that it is better than Growing NCL to all the databases used in the experiments. In section 4.2.3, we see that the training accuracy of the new MLPs for each one of the new data sets are lower than the accuracies of each MLP which compose the Fixed Size NCL. As we explain in that section, this is a reason about why Growing NCL has worse generalization than Fixed Size NCL. We also propose a modification in Growing NCL which might allow it to overcome this problem. Other comments about the differences in the generalization of the approaches are made in section 4.2.5.

4.2.2 Improvement in the Generalization

This section presents an analysis of the improvement in the generalization from the first to the last incremental steps. An improvement occurs when the generalization in the last incremental step is higher than the generalization in the first incremental step. A high improvement means that a good incremental ability is attained, while a decrease in the generalization is a signal of poor incremental learning ability. To support the analysis, the improvement obtained by SONG with 10 SGNTs is also presented.

Table 5 shows the generalization average in the first incremental step, in the last incremental step and the improvement in the generalization, respectively, considering the 30 executions of each approach for each database. The improvement was calculated as the generalization average in the last incremental step less the generalization average in the first incremental step. In this way, a high number indicates a high improvement. The best and the worse improvements for each one of the approaches are shown in bold font.

We can observe that there is no indication that the improvement is related to the difficulty of the database:

- For Fixed Size NCL, the best improvement occurred for the Letter database and the worse (even negative) for Vehicle.
- For Growing NCL, the best improvement occurred for the Optical Digits database and the worse for Adult.
- For SONG 10 SGNTs, the best improvement occurred for the Letter database and the worse for Mushroom.

We can also observe that:

Table 5: Improvement in the Generalization

(a) Fixed Size NCL

Database	Generalization Av.	Improvement
Letter	0.70362-0.81617	**0.11255**
Vehicle	0.77330-0.76898	**-0.00432**
Adult	0.83781-0.84294	0.00513
Optical Digits	0.83509-0.90464	0.06955
Mushroom	0.99233-0.99926	0.00693

(b) Growing NCL

Database	Generalization Av.	Improvement
Letter	0.66928-0.73193	0.06265
Vehicle	0.76173-0.77269	0.01096
Adult	0.83618-0.83823	**0.00205**
Optical Digits	0.76900-0.90027	**0.13127**
Mushroom	0.99086-0.99377	0.00291

(c) SONG 10 SGNTs

Database	Generalization Av.	Improvement
Letter	0.81255-0.94323	**0.13068**
Vehicle	0.59691-0.64398	0.04707
Adult	0.79657-0.79924	0.00268
Optical Digits	0.93517-0.96917	0.03400
Mushroom	0.99980-1.00000	**0.00020**

- The improvement of Fixed Size NCL was higher than Growing NCL in 3 databases and worse in 2.
- The improvement of Fixed Size NCL was higher than SONG 10 SGNTs in 3 databases and worse in 2.
- The improvement of Growing NCL was higher than SONG 10 SGNTs in 3 databases and worse in 2.

So, the improvement for the NCL approaches is higher than SONG 10 SGNTs in most of the databases and the improvement for Fixed Size NCL is higher than Growing NCL in most of the databases. However, the difference in the number of databases in which one approach has higher improvement than the other is always only one database. In this way, we can consider that the improvement for the NCL incremental approaches is comparable (in some cases higher and in others lower) with the improvement for SONG 10 SGNTs. This is an interesting observation, which again indicates that NCL is applicable to incremental learning.

Each approach has at least 2 databases in which the improvement is very low (less than 0.009). The improvement of Fixed Size NCL to Vehicle database was even negative, indicating a poor incremental behaviour of this approach to this database. To understand why an approach get a higher improvement than another and why a specific approach has a higher improvement when it learns some databases than the others, section 4.2.3 provides an analysis of the degradation of the accuracy on old training sets as new data sets are used for training.

4.2.3 Degradation

An important analysis to determine whether an approach is suffering catastrophic forgetting is the analysis of the degradation of the accuracy on old training sets as new data sets are used for training. To perform this analysis, the training sets previously learned are used to test the ensemble at each incremental step. It is important to notice that they are used only to test, and not to retrain the ensemble. The degradation of the accuracy of a particular training set is calculated as the accuracy average of this training set in the incremental step in which it was learned less the accuracy average in the last incremental step. A high decrease in the accuracy (degradation of the accuracy) of the training data sets from one incremental step to another indicates that the approach is suffering from catastrophic forgetting. As it will be shown, an execution in which there is too much catastrophic forgetting is not able to get high improvements in the generalization.

The analysis was performed with the Letter and the Vehicle data sets, which are the data sets in which Fixed Size NCL obtained the best and the worst improvement in the generalization, respectively. For both databases, SONG 10 SGNTs obtained a higher improvement in the generalization than the NCL approaches. The analysis can explain the differences in the improvement that occurs with Fixed Size NCL and SONG 10 SGNTs when compared to each other and when using different databases. An analysis was also made using the Optical Digits database, in order to allow comparisons with Learn++ and the evolutionary approach presented in section 3. The comparison with these approaches is explained in section 4.2.6.

The tables of this section show the accuracy averages of each one of the data sets (rows) in each of the incremental steps (columns). The last column presents the degradation. It is interesting to notice that the degradation of the generalization is equivalent to the opposite of the improvement of the generalization. However, we will use the word *degradation* only to the degradation of the accuracy of the training sets. Table 6 presents the results of the NCL incremental approaches to Letter, table 7 presents the results of SONG 10 SGNTs to Letter, table 8 presents the results of the NCL approaches to Vehicle, table 9 presents the results of SONG 10 SGNTs to Vehicle and table 10 presents the results of the NCL incremental approaches to Optical Digits.

Analysing Fixed Size NCL for Letter (table 6(a)), which is the database in which this approach obtained the highest improvement in the generalization, we can observe that the degradation is not so high (always lower than 0.05). We can also observe that the SONG 10 SGNTs' degradation to this database (table 7) is higher for the first three training sets, but is lower for all the others. So, SONG 10 SGNTs has lower degradations to this database than Fixed Size NCL and, as explained before, higher improvement in the generalization.

Analysing Fixed Size NCL for Vehicle (table 8(a)), which is the database in which this approach obtained the lowest improvement in the generalization, we can observe that the degradation is very high (always higher than 0.15). We can also observe that the SONG 10 SGNTs' degradation to this database (table 9) is lower for all incremental steps. As explained before, SONG 10 SGNTs has higher improvement in the generalization for this data set.

We can also observe that the SONG 10 SGNTs' degradation for Vehicle database is high in comparison with Letter database and the improvement in SONG 10 SGNTs' generalization rate for Vehicle was lower than for Letter.

Analysing Fixed Size NCL for Optical Digits (table 10(a)), which is the database in which this approach obtained an intermediate improvement in the generalization, we can observe that the degradation is usually higher than the degradation for Letter, but lower than the degradation for Vehicle.

For Fixed Size NCL, the degradation in the database with the worst improvement in the generalization (Vehicle) is higher than the degradation in the database with the highest improvement in the generalization (Letter). The degradation in the database with an intermediate improvement in the generalization has also an intermediate degradation. There is a similar behaviour for SONG 10 SGNTs. Besides, the degradation of SONG 10 SGNTs is lower than Fixed Size NCL, and SONG 10 SGNTs has a higher improvement than Fixed Size NCL in the generalization for both the databases. Thus, it is important to observe that the degradation is strongly related to the improvement in the generalization for Fixed Size NCL and SONG 10 SGNTs. A high degradation (which is related to catastrophic forgetting) causes a low improvement.

Analysing Growing NCL (tables 6(b), 8(b) and 10(b)), we can observe a very good behaviour in relation to catastrophic forgetting, as expected. The degradation is very low and, in many cases, negative. A negative degradation means that the accuracy in the old training sets is increasing, which is a good characteristic to incremental learning.

For Vehicle and Optical Digits, we can observe that the improvement of Growing NCL in the generalization is higher than the improvement of Fixed Size NCL. However, for Letter, the improvement is worse, even the degradation being lower. The problem of Growing NCL is that only one MLP is inserted in the ensemble to be trained with each new data set, while in Fixed Size NCL all the MLPs of the ensemble are used to learn the new data set. The short number of MLPs to each new data set and the fact that each new MLP has not received any previous training has a bad influence in Growing NCL results. We can observe in tables 6, 8 and 10 that the accuracy of each new training set is always lower than the accuracy obtained by Fixed Size NCL. The higher accuracy on the first training set (in the first incremental step) is related to the fact that the Fixed Size NCL uses all the MLPs of the ensemble to learn the data, while Growing NCL uses only one. We can observe that the accuracies on the training sets of the other incremental steps of Fixed Size NCL tend to increase more than the accuracies of Growing NCL, as in Fixed Size NCL the MLPs used to learn new data already received previous training with data that can have similarities with the new incoming data.

A possible solution to Growing NCL problems is to increase the size of the ensemble in more than one MLP to each new data set. The problem of this solution is that the size of the growing ensemble could become very large. Another possible solution to Growing NCL problems is to set the initial weights of the new MLP as the weights of the last previously trained MLP. A third possible solution would be to use not only the new MLP, but also all the MLPs that were already inserted in the ensemble to learn the new data set. This solution could prejudice the Growing NCL good behaviour to catastrophic forgetting.

Table 6: Degradation - NCL - Letter Database

(a) Fixed Size NCL

Train Set \ Inc	1	2	3	4	5	6	7	8	9	Degradation
1	0.77339	0.78360	0.79780	0.80185	0.79377	0.79547	0.80845	0.81618	0.81278	-0.03938
2		0.82343	0.79432	0.79547	0.79442	0.79905	0.80948	0.81348	0.81428	0.00915
3			0.84909	0.82361	0.81690	0.82798	0.83403	0.83388	0.83413	0.01495
4				0.84431	0.80020	0.80453	0.80955	0.81643	0.81685	0.02746
5					0.85419	0.81485	0.82481	0.81603	0.81595	0.03823
6						0.86257	0.82908	0.82678	0.82361	0.03896
7							0.85991	0.82646	0.82303	0.03688
8								0.86652	0.83143	0.03508
9									0.86237	0.00000
Generalization	0.70362	0.77863	0.78808	0.79643	0.79938	0.80467	0.81482	0.81927	0.81617	-0.11255

(b) Growing NCL

Train Set \ Inc	1	2	3	4	5	6	7	8	9	Degradation
1	0.73256	0.71410	0.73906	0.73743	0.73268	0.73613	0.73808	0.73661	0.73493	-0.00238
2		0.70260	0.72578	0.72396	0.71960	0.72208	0.72656	0.72611	0.72811	-0.02551
3			0.74404	0.74146	0.73718	0.74241	0.74406	0.74229	0.74231	0.00173
4				0.72636	0.72196	0.72666	0.73028	0.73061	0.73031	-0.00395
5					0.73436	0.73798	0.73878	0.73666	0.73633	-0.00198
6						0.73988	0.74239	0.74061	0.73986	0.00003
7							0.73011	0.72871	0.72786	0.00225
8								0.74299	0.74336	-0.00038
9									0.73326	0.00000
Generalization	0.66928	0.68512	0.71963	0.72197	0.71873	0.72395	0.72997	0.72995	0.73193	-0.06265

Table 7: Degradation - SONG 10 SGNTs - Letter Database

Train Set \ Inc	1	2	3	4	5	6	7	8	9	Degradation
1	0.99427	0.96732	0.95712	0.95553	0.95700	0.95853	0.96183	0.96317	0.96212	0.03215
2		0.99955	0.99093	0.98432	0.97940	0.97828	0.97263	0.97003	0.97107	0.02848
3			0.99973	0.99513	0.99235	0.98790	0.98457	0.98470	0.98215	0.01758
4				0.99982	0.99833	0.99387	0.99122	0.98937	0.98698	0.01283
5					0.99988	0.99835	0.99610	0.99483	0.99267	0.00722
6						0.99993	0.99918	0.99810	0.99690	0.00303
7							0.99995	0.99892	0.99820	0.00175
8								0.99993	0.99912	0.00082
9									0.99995	0.00000
Generalization	0.81255	0.86427	0.89037	0.90720	0.91973	0.92530	0.93165	0.93898	0.94323	-0.13068

Table 8: Degradation - NCL - Vehicle Database

(a) Fixed Size NCL

Train Set \ Inc	1	2	3	Degradation
1	0.98286	0.86405	0.82167	0.16119
2		0.97024	0.77381	0.19643
3			0.98238	0.00000
Generalization	0.77330	0.75926	0.76898	0.00432

(b) Growing NCL

Train Set \ Inc	1	2	3	Degradation
1	0.98071	0.92929	0.88405	0.09667
2		0.80810	0.83262	-0.02452
3			0.79310	0.00000
Generalization	0.76173	0.72500	0.77269	-0.01096

Table 9: Degradation - SONG 10 SGNTs - Vehicle Database

Train Set \ Inc	1	2	3	Degradation
1	0.98651	0.91381	0.86190	0.12460
2		0.99794	0.97651	0.02143
3			0.99984	0.00000
Generalization	0.59691	0.64089	0.64398	-0.04707

Table 10: Degradation - NCL - Optical Digits Database

(a) Fixed Size NCL

Train Set \ Inc	1	2	3	4	5	6	Degradation
1	1.00000	0.95940	0.95815	0.93684	0.93659	0.93534	0.06466
2		0.99825	0.96617	0.94912	0.95088	0.90627	0.09198
3			0.99950	0.98070	0.97469	0.96992	0.02957
4				0.99850	0.95363	0.95439	0.04411
5					1.00000	0.96140	0.03860
6						1.00000	0.00000
Generalization	0.83509	0.89333	0.90543	0.91100	0.92379	0.90464	-0.06955

(b) Growing NCL

Train Set \ Inc	1	2	3	4	5	6	Degradation
1	0.99248	0.90927	0.92030	0.91429	0.92782	0.91203	0.08045
2		0.87018	0.91504	0.90276	0.93158	0.91353	-0.04336
3			0.94110	0.95013	0.96190	0.95414	-0.01303
4				0.89649	0.91429	0.91830	-0.02180
5					0.93659	0.93484	0.00175
6						0.96817	0.00000
Generalization	0.76900	0.76743	0.83667	0.87176	0.89698	0.90027	-0.13127

4.2.4 Best Networks of the Ensemble

This section presents an analysis of the capacity that some MLPs of the Fixed Size NCL ensemble have to adapt better than the others for new data. In order to do that, a single execution of Fixed Size NCL to each one of the databases was chosen. The execution was the one which obtained the intermediate generalization in the last incremental step. Each incremental step of each one of the chosen executions was analysed to determine the best MLP of the ensemble in that incremental step, which is considered the MLP with the highest generalization. The change in the best MLP from one incremental step to another shows that different MLPs are adapting differently to new data and some are adapting better than the others. This is an important behaviour to incremental learning and one of the advantages of using ensembles to perform incremental learning, as it was briefly explained in section 1.

Table 11 shows the best MLP to each incremental step (Inc. Step), for all databases, and the percentage of repetition (Rep.) of the best MLP in consecutive incremental steps. Each MLP of the ensemble is identified by a number, from 1 to 10. When more than 1 MLP has the same generalization which is the highest (the best), all the best MLPs are listed and separated with comma.

Table 11: Best MLP of Each Incremental Step

Inc. Step	Letter Best MLP	Vehicle Best MLP	Adult Best MLP	Opt. Digits Best MLP	Mush. Best MLP
1	2	9	6	5	7
2	7	7	5	4	1
3	7	3	8	4	4, 7
4	8	-	1	4	1, 4, 7
5	8	-	10	4	1
6	7	-	1	2	1, 6
7	3	-	1	-	2, 6, 7, 8
8	2	-	9	-	1, 2, 7
9	4	-	8	-	1, 2, 6, 7
Rep.	22% (2/9)	0% (0/3)	11% (1/9)	50% (3/6)	66% (6/9)

We can see that the best MLP of the ensemble from one incremental step to another changes. This shows that, as new data sets come, different MLPs of the ensemble can adapt better. Only for the easiest data sets there was a high number of consecutive incremental steps in which the best MLP was the same, which is a reasonable behaviour. This shows one of the advantages of using an ensemble with diverse neural networks to incremental learning. When a single MLP is utilized to perform incremental learning, it can have a good adaptation to a particular data set, but not to another, decreasing the generalization of the approach. Using an ensemble, the neural networks that adapt better and faster to new data can compensate the misclassifications of the others.

4.2.5 Intersection of the Correct Response Sets

The execution of each approach to each database which has the intermediate generalization in the last incremental step was used to perform an analysis of the intersection of the correct response sets. The analysis is performed with Fixed Size NCL, Growing NCL and SONG 10 SGNTs. The notion of correct response sets and their intersections was introduced by Liu and Yao (1999a). The correct response set S_i of the individual neural network i on the testing set consists of all patterns in the testing set which are classified correctly by the individual network i. In (Liu and Yao; 1999a), Ω_i denotes the size of the set S_i and $\Omega_{i_1,i_2,...,i_k}$ denotes the size of the set $S_{i_1} \cap S_{i_2} \cap ... \cap S_{i_k}$. In the analysis performed in this section, Ω denotes the percentage of test patterns that is in the intersection, instead of the number of patterns that is in the intersection.

A high Ω indicates that the neural networks participating in the intersection are more similar than a low Ω. It is expected that an ensemble has better generalization than its individual members when its members are diverse (different from each other) and have accuracy higher than 0.5. So, it is interesting to analyse whether the approaches with the best generalizations have some relation with a higher or lower Ω among all its neural networks. As the definition of correct response sets is based on the correctly classified patterns for each ensemble member, an Ω value lower than 1 indicates that the ensemble members participating in the intersection perform differently on the same training data. As it was briefly commented in section 1, the difference among the ensemble members is important for that the neural networks which have a better adaptation to new incoming data can overcome the problem of the ones which have a bad adaptation.

In this section, first, the intersections between the correct response set of the best and the second best neural networks of each incremental step and the intersection between the best and the worst neural networks of each incremental step for Fixed Size NCL and SONG 10 SGNTs are analysed. This analysis is performed to give some ideas about the relationship between a high Ω with similar neural networks. Tables 12, 13, 14, 15 and 16 show these Ω values for each of the databases.

We can observe that $\Omega_{best,2^{nd}best}$ is usually higher than $\Omega_{best,worst}$, indicating that MLPs or SGNTs with more similar generalization have also higher Ω. We can also observe that, as new trainings are received by the components of the ensembles, usually both $\Omega_{best,2^{nd}best}$ and $\Omega_{best,worst}$ tend to increase. As more training is received by the MLPs or SGNTs, their generalization become more similar to each other, also indicating that a higher Ω is related to more similar MLPs or SGNTs in the ensemble.

However, it is interesting to notice that the increase in $\Omega_{best,2^{nd}best}$ and $\Omega_{best,worst}$ values is not

Table 12: Intersection of the Correct Response Sets (Letter) - $\Omega_{best,2^{nd}best}$ and $\Omega_{best,worst}$

Inc.	Fixed Size NCL		SONG 10 SGNTs	
Step	$\Omega_{best,2^{nd}best}$	$\Omega_{best,worst}$	$\Omega_{best,2^{nd}best}$	$\Omega_{best,worst}$
1	0.60050	0.56650	0.62500	0.61200
2	0.65200	0.62750	0.70600	0.69200
3	0.65600	0.64100	0.73900	0.73300
4	0.68450	0.63150	0.84700	0.76500
5	0.67250	0.63250	0.78900	0.78800
6	0.68050	0.64050	0.80900	0.79900
7	0.66800	0.65000	0.81300	0.80400
8	0.67800	0.65850	0.83100	0.82200
9	0.66000	0.64250	0.83600	0.82100
Av.	0.66133	0.63228	0.77722	0.75956

Table 13: Intersection of the Correct Response Sets (Vehicle) - $\Omega_{best,2^{nd}best}$ and $\Omega_{best,worst}$

Inc.	Fixed Size NCL		SONG 10 SGNTs	
Step	$\Omega_{best,2^{nd}best}$	$\Omega_{best,worst}$	$\Omega_{best,2^{nd}best}$	$\Omega_{best,worst}$
1	0.68056	0.65741	0.44400	0.41700
2	0.68981	0.62037	0.48600	0.47200
3	0.63889	0.62500	0.66200	0.47200
Av.	0.66975	0.63426	0.53067	0.45367

steady. There are some incremental steps in which $\Omega_{best,2^{nd}best}$ and/or $\Omega_{best,worst}$ have considerable decrease in relation to the previous incremental step. Regarding 0.01 as a considerable decrease, we can observe that Fixed Size NCL has decreases in 4 out of 5 databases, while SONG has decrease only in 1 of the incremental steps of 1 of the databases. This represents a success of NCL to create and maintain diversity in the ensemble. An extreme case is the Vehicle database, in which the last incremental step has even lower $\Omega_{best,2^{nd}best}$ and $\Omega_{best,worst}$ values than the first incremental step when Fixed Size NCL is used.

Tables 17, 18 and 19 present the intersection among the correct response sets of all the MLPs/SGNTs of the ensemble (Ω_{all}), for each incremental step. We can observe that all Ω_{all} values are lower and 1, indicating that the neural networks in the ensemble perform differently even being trained with the same data. However, we can also observe that, for the databases in which Fixed Size NCL has higher generalization than SONG 10 SGNTs (Vehicle and Adult), the intersections of Fixed Size NCL are always greater than the intersections of SONG 10 SGNTs. For the databases in which SONG 10 SGNTs has higher generalization than Fixed Size NCL (Letter, Optical Digits and Mushroom), the intersections of Fixed Size NCL are always lower than the intersections of SONG 10 SGNTs (except the first incremental step of Letter and Mushroom). This fact seems to be contradictory to the fact that ensembles composed by diverse members have higher generalization. Nevertheless, a too low Ω_{all} indicates that the components of the ensemble are too different from each other and may not

Table 14: Intersection of the Correct Response Sets (Adult) - $\Omega_{best,2^{nd}best}$ and $\Omega_{best,worst}$

Inc.	Fixed Size NCL		SONG 10 SGNTs	
Step	$\Omega_{best,2^{nd}best}$	$\Omega_{best,worst}$	$\Omega_{best,2^{nd}best}$	$\Omega_{best,worst}$
1	0.82224	0.80942	0.67900	0.66600
2	0.80699	0.77294	0.68700	0.68100
3	0.81671	0.77161	0.70100	0.69800
4	0.82268	0.79262	0.69700	0.69400
5	0.83838	0.79239	0.69600	0.69000
6	0.82799	0.80522	0.68800	0.68800
7	0.82467	0.78598	0.69300	0.68400
8	0.82821	0.80544	0.69400	0.68800
9	0.82954	0.80721	0.78400	0.69700
Av.	0.82416	0.79365	0.70211	0.68733

Table 15: Intersection of the Correct Response Sets (Optical Digits) - $\Omega_{best,2^{nd}best}$ and $\Omega_{best,worst}$

Inc.	Fixed Size NCL		SONG 10 SGNTs	
Step	$\Omega_{best,2^{nd}best}$	$\Omega_{best,worst}$	$\Omega_{best,2^{nd}best}$	$\Omega_{best,worst}$
1	0.69887	0.57308	0.88900	0.88200
2	0.79163	0.72715	0.91400	0.90600
3	0.81833	0.75317	0.92600	0.91600
4	0.85611	0.77805	0.93000	0.92200
5	0.87647	0.71154	0.93100	0.92700
6	0.84005	0.72647	0.93900	0.93100
Av.	0.81357	0.71158	0.92150	0.91400

Table 16: Intersection of the Correct Response Sets (Mushroom) - $\Omega_{best,2^{nd}best}$ and $\Omega_{best,worst}$

Inc.	Fixed Size NCL		SONG 10 SGNTs	
Step	$\Omega_{best,2^{nd}best}$	$\Omega_{best,worst}$	$\Omega_{best,2^{nd}best}$	$\Omega_{best,worst}$
1	0.99262	0.98278	1.00000	0.99100
2	0.99631	0.99016	1.00000	0.99800
3	0.99754	0.98647	0.99900	0.99100
4	0.99631	0.98770	0.99900	0.99900
5	0.99877	0.99016	1.00000	1.00000
6	0.99877	0.99139	1.00000	1.00000
7	0.99877	0.99385	1.00000	1.00000
8	0.99877	0.99508	1.00000	1.00000
9	0.99754	0.99508	1.00000	1.00000
Av.	0.99727	0.99030	0.99978	0.99767

contribute to a good result from the ensemble. Thus, further analysis is required to check the relationship between Ω_{all} and generalization.

Table 17: Intersection of the Correct Response Sets (Letter and Vehicle) - Ω_{all}

	Letter		Vehicle	
Inc. Step	Fixed Size NCL Ω_{all}	SONG 10 SGNTs Ω_{all}	Fixed Size NCL Ω_{all}	SONG 10 SGNTs Ω_{all}
1	0.44250	0.34000	0.53704	0.26900
2	0.42900	0.46500	0.42593	0.27800
3	0.46950	0.51600	0.35648	0.23600
4	0.46900	0.55600	-	-
5	0.46950	0.57600	-	-
6	0.48800	0.60300	-	-
7	0.45850	0.61500	-	-
8	0.45500	0.62700	-	-
9	0.45700	0.64100	-	-

Table 18: Intersection of the Correct Response Sets (Adult) - Ω_{all}

	Adult	
Inc. Step	Fixed Size NCL Ω_{all}	SONG 10 SGNTs Ω_{all}
1	0.77824	0.45900
2	0.74066	0.49100
3	0.71391	0.51400
4	0.75702	0.51800
5	0.74707	0.52200
6	0.76476	0.51000
7	0.73049	0.51500
8	0.77382	0.51000
9	0.77625	0.51400

Table 20 shows Ω_{all} to each incremental step of the Growing NCL. The first incremental step listed in the table is the second incremental step, as in the first there is only one MLP in the ensemble. It is possible to observe that Ω_{all} always decreases from one incremental step to another. This is a reasonable behaviour, as, from one incremental step to another, a new MLP is added to the ensemble.

The Growing NCL Ω_{all} is almost always higher than the Fixed Size NCL Ω_{all}, for the second and third incremental steps. For the other incremental steps, Ω_{all} of Growing NCL is always lower than the Ω_{all} of Fixed Size NCL. Again, a relation between a low Ω_{all} and a worse generalization could be indicated. However, it is important to remember that the worse generalizations of Growing NCL in relation to Fixed Size NCL have other causes, as explained mainly in section

Table 19: Intersection of the Correct Response Sets (Optical Digits and Mushroom) - Ω_{all}

	Optical Digits		Mushroom	
Inc. Step	Fixed Size NCL Ω_{all}	SONG 10 SGNTs Ω_{all}	Fixed Size NCL Ω_{all}	SONG 10 SGNTs Ω_{all}
1	0.37262	0.79400	0.44250	0.34000
2	0.53846	0.82500	0.42900	0.46500
3	0.52670	0.84200	0.46950	0.51600
4	0.54751	0.85100	0.46900	0.55600
5	0.66290	0.85500	0.46950	0.57600
6	0.69434	0.85600	0.48800	0.60300
7	-	-	0.45850	0.61500
8	-	-	0.45500	0.62700
9	-	-	0.45700	0.64100

4.2.3. Thus, the relationship between Ω_{all} and the generalization of the ensembles needs further studies.

Section 5 presents some discussion about the relationship between diversity and generalization.

Table 20: Intersection of the All Correct Response Sets For Growing NCL - Ω_{all}

Inc. Step	Letter	Vehicle	Adult	Opt. Digits	Mush.
2	0.58900	0.57870	0.75039	0.67964	0.98401
3	0.48650	0.46759	0.73447	0.55814	0.93850
4	0.36350	-	0.72761	0.46176	0.92497
5	0.28950	-	0.71656	0.43484	0.92497
6	0.13600	-	0.69246	0.37127	0.83026
7	0.12900	-	0.67698	-	0.83026
8	0.12850	-	0.66947	-	0.83026
9	0.11250	-	0.62945	-	0.78106

4.2.6 Comparison With Other Approaches

In this section, a comparison between the NCL incremental approaches with Learn++ and Seipone and Bullinaria (2005)'s evolutionary approach is presented.

Comparing Fixed Size NCL (tables 8(a) and 10(a)) and Learn++ (tables 21 and 22(a)), Fixed Size NCL has:

- Higher generalization average in the first incremental steps, but lower in the last incremental steps, for Optical Digits.
- Lower generalization average in all incremental steps, for Vehicle.

- Higher degradations, for Optical Digits and Vehicle.
- Lower generalization improvement, for Optical Digits and Vehicle.

Comparing Growing NCL (tables 8(b) and 10(b)) and Learn++ (tables 21 and 22(a)), Growing NCL has:

- Lower generalization averages in all incremental steps, for Optical Digits and Vehicle.
- Comparable degradations, for Optical Digits.
- Lower degradations, for Vehicle.
- Higher generalization improvement, for Optical Digits.
- Lower generalization improvement, for Vehicle.

It is important to observe that an ensemble of 30 MLPs is constructed by Learn++ to each new data set. In this way, the ensemble size of Learn++ is much higher than the size of the NCL incremental approaches, that have at most 10 MLPs. This fact may explain why Learn++ got better generalization than the incremental approaches. It is also important to remember that Growing NCL has problems that make it have a low generalization improvement even when the degradations are low.

Comparing the Fixed Size NCL (table 10(a)) with the evolutionary approach (table 22(b)), it is possible to observe that Fixed Size NCL has:

- Lower generalization in all the incremental steps.
- Higher degradations.
- Higher generalization improvement.

The higher generalization improvement of Fixed Size NCL even with higher degradations can be related to the fact that the evolutionary approach already has a very high generalization in the first incremental step in comparison with the NCL incremental approaches and Learn++.

Comparing the Growing NCL (table 10(b)) with the evolutionary approach (table 22(b)), it is possible to observe that Growing NCL has:

- Lower generalization in all the incremental steps.
- Lower degradations in all but the first incremental step.

- Higher generalization improvement.

It is important to remember that the evolutionary approach considers all the data sets in each generation, being biased to the presentation of these data sets.

Table 21: Degradation - Learn++ - Vehicle Database

Train Set \ Inc	1	2	3	Degradation
1	0.930	0.820	0.790	0.140
2		0.860	0.780	0.080
3			0.100	0.000
Generalization	0.780	0.804	0.83000	-0.05

Table 22: Degradation - Learn++ and Evolutionary Approach - Optical Digits Database

(a) Learn++

Train Set \ Inc	1	2	3	4	5	6	Degradation
1	0.940	0.940	0.940	0.930	0.930	0.930	0.010
2		0.935	0.940	0.940	0.940	0.930	0.005
3			0.950	0.940	0.940	0.940	0.010
4				0.935	0.940	0.940	-0.005
5					0.950	0.950	0.000
6						0.950	0.000
Generalization	0.820	0.847	0.897	0.917	0.922	0.927	-0.107

(b) Evolutionary Approach

Train Set \ Inc	1	2	3	4	5	6	Degradation
1	1.000	0.988	0.984	0.981	0.979	0.980	0.020
2		1.000	0.989	0.983	0.980	0.978	0.022
3			1.000	0.990	0.985	0.983	0.017
4				1.000	0.991	0.985	0.015
5					1.000	0.989	0.011
6						1.000	0.000
Generalization	0.914	0.929	0.936	0.939	0.942	0.944	-0.030

5 Discussion About Diversity, Accuracy and Generalization

This section presents some discussions about the relationship between diversity and generalization. Diversity among the ensemble members is very important to produce a successful ensemble (Dietterich; 1997) and empirical results have been showing a positive correlation between accuracy of the ensemble and diversity among its members (Dietterich; 2000; Kuncheva and Whitaker; 2003). However, considering the analysis of the intersection of the correct response sets explained in section 4.2.5, a low intersection (high diversity) not necessarily leads to good generalization. A more rigorous and comprehensive study of various diversity measures in ensembles is outside the scope of this paper and can be found elsewhere (Tang et al.; 2006).

In addition to the study of diversity, some authors have studied the concept of margin to analyse the success of ensembles (Schapire et al.; 1998; Breiman; 2001; Rätsch et al.; 2001). According to Schapire et al. (1998), "the margin of an example is simply the difference between the number of correct votes and the maximum number of votes received by any incorrect label". They proved that larger margins on the training set result in improved generalization error bound. Rätsch et al. (2001) explained the good generalization obtained by ensembles through the analysis of the minimum margin that they can achieve. An ensemble with the largest minimum margin will have best generalization error bound.

Tang et al. (2006) studied the relationship between diversity and margins of an ensemble. Six different diversity measures for classifiers are analysed. They conclude that when the average classification accuracy of the ensemble members on the training data is considered a constant, maximizing the diversity is equivalent to maximizing the minimum margin of the ensemble on the training instances. In order to maximize the minimum margin of an ensemble, it is necessary to satisfy the following equation, called uniformity condition:

$$l_i = L(1 - P) \quad \forall i \ , \tag{15}$$

where L is the number of ensemble members, P is the average classification accuracy of the ensemble members on the training data and l_i is the number of ensemble members that misclassify the training instance i, when the ensemble members are unweighted.

However, l_i is a discrete value, while $L(1 - P)$ is a continuous value. So, the uniformity condition usually cannot be satisfied and the maximum diversity is usually not achievable. Besides, the minimum margin of an ensemble does not monotonically increase with respect to diversity (Tang et al.; 2006). Hence, enlarging diversity is not exactly the same as enlarging the minimum margin. Based on that, they conclude that large diversity may not always correspond to better generalization. The results obtained in section 4.2.5 are compatible with this analysis.

It is usually affirmed in the literature that there is a trade-off between accuracy and diversity, meaning that lower accuracy may correspond to higher diversity. However, the relationship between accuracy and diversity is not straightforward and this affirmation is not reasonable. According to Tang et al. (2006), all the diversity measures that they analysed can be formulated as $diversity = a - (bP + c\sum_{i=1}^{N} l_i^2)$, where a, b and c are constants and N is the number of training instances. So, the relationship between diversity and P is influenced by the term $\sum_{i=1}^{N} l_i^2$. The experiments performed by the authors indicate that there is a strong negative correlation between P and $\sum_{i=1}^{N} l_i^2$. So, a lower P is not likely to result in a lower $(bP + c\sum_{i=1}^{N} l_i^2)$ and hence may not correspond to a higher diversity.

As we can see, the relationship between diversity, accuracy on the training set and generalization is complex. It is not possible to affirm that higher diversity can lead to better generalization and that there is a trade-off between diversity and accuracy.

6 Conclusions

This paper investigates the use of negative correlation in incremental learning, determining its strong and weak points to incremental learning. NCL is a successful approach to construct neural network ensembles. In off-line mode, it has shown to outperform other ensemble learning methods (Islam et al.; 2003; Wang et al.; 2004; Chandra and Yao; 2006). It directly encourages diversity by making the learning of an ensemble member be influenced by the learning of the other ensemble members. The difference among the neural networks that compose an ensemble is a desirable feature to perform incremental learning, for some of the neural networks can be able to adapt faster and better to new data than the others, possibly overcoming the problems of the ones which have bad adaptation. So, NCL is a potentially powerful approach to incremental learning.

Two different approaches to use negative correlation in incremental learning (called Fixed Size NCL and Growing NCL) are presented and analysed. To support the analysis, NCL is compared with SONG, which is a successful approach to incremental learning.

The analysis shows that it is possible to use negative correlation in incremental learning, although each approach also has its weakness. Therefore, it would be interesting to develop an approach which combines the advantages of both Fixed Size and Growing NCL as a future work. Some of the identified strong points and weakness of the approaches are:

- Fixed Size NCL

Strong points - The analysis shows that different networks of the ensemble adapt differently to new data. Some can adapt better than the others and the best neural networks are not the same from one incremental step to another. This is one of the advantages of using negative correlation in incremental learning, as neural networks which adapt better can make the ensemble overcome the problem of the neural networks which could not have a good adaptation to new data.

Weakness - This approach suffers more catastrophic forgetting than Growing NCL.

- Growing NCL

  Strong points - The analysis shows that it is possible to use Growing NCL to overcome catastrophic forgetting, an important problem related to incremental learning.

  Weakness - This approach has low generalization in comparison with Fixed Size NCL and do not take advantage of using more than one neural network of the ensemble to learn new data.

It is important to observe that the improvement in the generalization after training with new data for both the approaches is comparable with other approaches specifically developed to incremental learning existent in the literature. This result is very encouraging, indicating that new approaches combining Fixed Size and Growing NCL may overcome the improvements achieved by the current approaches and obtain even better generalization.

Three different ways to combine both the NCL incremental approaches are proposed as future works. One of them is to include more than only one MLP to learn each new data set in Growing NCL. Another way is to initialise the weights of the new MLP which is inserted in Growing NCL with the weights learned by the last previously inserted MLP. A third way is to train all the MLPs of Growing NCL with the new data set, not only the new MLP which is inserted when the new data set is presented.

Further investigation about the relationship between the size of the intersection of the correct response sets (Ω) and the generalization obtained by the incremental approaches is also left as a future work, as well as the analysis of the behaviour of the NCL incremental approaches to the presentation of new classes in incoming data sets.

In conclusion, the analyses show that Fixed Size and Growing NCL are comparable with other approaches which were **specifically** designed to incremental learning and have strong points which combined may be able to outperform the current approaches existent in the literature. Thus, the study presented in this work reveals encouraging results with negative correlation in incremental learning and shows that NCL is a promising approach to incremental learning.

Acknowledgements

The first author would like to thank the United Kingdom Government and the School of Computer Science of the University of Birmingham for the financial support in the form of an Overseas Research Students Award (ORSAS) and a School Research Scholarship. The authors are grateful to the guest editor, Professor Bogdan Gabrys, and anonymous referees for their valuable comments, which have helped to improve the quality of this paper.

References

Adamczak, R., Duch, W. and Jankowski, N. (1997). New developments in the feature space mapping model, *Proceedings of the Third Conference on Neural Networks and Their Applications*, Kule, Poland, pp. 65–70.

Breiman, L. (1996). Bagging predictors, *Machine Learning* **24**(2): 123–140.

Breiman, L. (2001). Random forests, *Machine Learning* **45**(1): 5–32.

Brown, G. (2004). *Diversity in Neural Network Ensembles*, PhD thesis, School of Computer Science, The University of Birmingham, Birmingham, UK.
URL: *http://www.cs.man.ac.uk/∼gbrown/research.php*

Brown, G., Wyatt, J. L. and Tiño, P. (2005). Managing diversity in regression ensembles, *Journal of Machine Learning Research* **6**: 1621–1650.

Carpenter, G. A., Grossberg, S., Markuzon, N. and Reynolds, J. H. (1992). Fuzzy ARTMAP: A neural network architecture for incremental supervised learning of analog multidimensional maps, *IEEE Transactions on Neural Networks* **3**: 698–713.

Carpenter, G. A., Grossberg, S. and Reynolds, J. H. (1991). ARTMAP: Supervied real-time learning and classification of nonstationary data by a self organizing neural network, *Neural Networks* **4**(5): 565–588.

Chandra, A., Chen, H. and Yao, X. (2006). *Multi-objective Machine Learning*, Springer-Verlag, chapter Trade-off between diversity and accuracy in ensemble generation, pp. 429–464.

Chandra, A. and Yao, X. (2006). Evolving hybrid ensembles of learning machines for better generalisation, *Neurocomputing* **69**: 686–700.

Dietterich, T. (2000). An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting and randomization, *Machine Learning* **40**(2): 1–22.

Dietterich, T. G. (1997). Machine learning research: Four current directions, *AI Magazine* **18**: 97–136.

Dietterich, T. G. (1998). Approximate statistical tests for comparing supervised classification learning algorithms, *Neural Computation* **10**: 1895–1923.

Eiben, A. E. and Smith, J. E. (2003). *Introduction to Evolutionary Computing*, New York: Springer-Verlag Berlin Heidelberg.

Freund, Y. and Schapire, R. (1997). A decision-theoretic generalization of on-line learning and an application to boosting, *Journal of Computer and System Sciences* **55**(1): 119–139.

Inoue, H. and Narihisa, H. (2000). Improving generalization ability of self-generating neural networks through ensemble averaging, *Proceedings of the Fourth Pacific-Asia Conference on Knowledge Discovery and Data Mining (LNAI 1805)*, Kyoto, Japan, pp. 177–180.

Inoue, H. and Narihisa, H. (2003). Effective pruning method for a multiple classifier system based on self-generating neural networks, *Proceedings of the 2003 Joint International Conference (ICANN/ICONIP'03 - LNCS 2714)*, Istanbul, Turkey, pp. 11–18.

Inoue, H. and Narihisa, H. (2005). Self-organizing neural grove and its applications, *Proceedings of the 2005 International Joint Conference on Neural Networks (IJCNN'05)*, Montreal, Canada, pp. 1205–1210.

Islam, M. M., Yao, X. and Murase, K. (2003). A constructive algorithm for training cooperative neural network ensembles, *IEEE Transactions on Neural Networks* **14**(4): 820–834.

Kasabov, N. (2001). Evolving fuzzy neural networks for supervised/unsupervised online knowledge-based learning, *IEEE Transactions on Systems, Man and Cybernetics - Part B: Cybernetics* **31**(6): 902–918.

Kohonen, T. (1995). *Self-Organizing Maps*, Springer-Verlag, Berlin.

Kuncheva, L. and Whitaker, C. (2003). Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy, *Machine Learning* **51**(2): 181–207.

Larose, D. T. (2004). *Discovering Knowledge in Data: An Introduction to Data Mining*, Wiley-Interscience.

Liu, Y. and Yao, X. (1999a). Ensemble learning via negative correlation, *Neural Networks* **12**: 1399–1404.

Liu, Y. and Yao, X. (1999b). Simultaneous training of negatively correlated neural networks in an ensemble, *IEEE Transactions on Systems, Man and Cybernetics Part B - Cybernetics* **29**(6): 716–725.

Newman, D., Hettich, S., Blake, C. and Merz, C. (1998). UCI repository of machine learning databases.
URL: *http://www.ics.uci.edu/∼mlearn/MLRepository.html*

Polikar, R., Udpa, L., Udpa, S. S. and Honavar, V. (2001). Learn++: An incremental learning algorithm for supervised neural networks, *IEEE Transactions on Systems, Man, and Cybernetics - Part C: Applications and Reviews* **31**(4): 497–508.

Prechelt, L. (1994). PROBEN1 - a set of neural network benchmark problems and benchmarking rules, *Technical Report 21/94*, Fakultt fr Informatik, Universitt Karlsruhe, Karlsruhe, Germany.

Rätsch, G., Onoda, T. and Müller, K.-R. (2001). Soft margins for AdaBoost, *Machine Learning* **42**(3): 287–320.

Rumelhart, D. E., Hinton, G. E. and Williams, R. J. (1986). Learning internal representations by error propagation, *Parallel Distributed Processing: Explorations in the Microstructures of Cognition* **I**: 318–362.

Schapire, R. (1990). Strength of weak learning, *Machine Learning* **5**: 197–227.

Schapire, R. E., Freund, Y., Bartlett, P. L. and Lee, W. S. (1998). Boosting the margin: A new explanation for the effectiveness of voting methods, *Annals of Statistics* **26**(5): 1651–1686.

Seipone, T. and Bullinaria, J. (2005). Evolving improved incremental learning schemes for neural network systems, *Proceedings of the 2005 IEEE Congress on Evolutionary Computing (CEC'2005)*, Piscataway, NJ, pp. 273–280.

Tang, E. K., Suganthan, P. N. and Yao, X. (2006). An analysis of diversity measures, *Machine Learning* **62**(1): 247–271.

Wang, Z., Yao, X. and Xu, Y. (2004). An improved constructive neural network ensemble approach to medical diagnoses, *Proceedings of the Fifth International Conference on Intelligent Data Engineering and Automated Learning (IDEAL'04), Lecture Notes in Computer Science*, Vol. 3177, Springer, Exeter, UK, pp. 572–577.

Wen, W. X., Jennings, A. and Liu, H. (1992). Learning a neural tree, *Proceedings of the 1992 International Joint Conference on Neural Networks (IJCNN'92)*, Vol. 2, Beijing, China, pp. 751–756.

Witten, I. H. and Frank, E. (2000). *Data Mining - Pratical Machine Learning Tools and Techniques with Java Implementations*, Morgan Kaufmann Publishers, San Francisco.

Zanchettin, C., Minku, F. L. and Ludermir, T. B. (2005). Design of experiments in neuro-fuzzy systems, *Proceedings of the 5th International Conference on Hybrid Intelligent Systems, HIS'2005*, Rio de Janeiro, Brasil, pp. 218–223.