On Combining Artificial Neural Nets

Amanda J.C. Sharkey
Department of Computer Science,
University of Sheffield, U.K.

Abstract

This paper reviews research on combining artificial neural nets, and provides an overview of, and an introduction to, the papers contained this Special Issue, and its companion (*Connection Science*, **9**, 1). Two main approaches, ensemble-based, and modular, are identified and considered. An ensemble, or committee, is made up of a set of nets, each of which is a general function approximator. The members of the ensemble are combined in order to obtain better generalisation performance than would be achieved by any of the individual nets. The main issues considered here under the heading of ensemble-based approaches, are (a) how to combine the outputs of the ensemble members (b) how to create candidate ensemble members and (c) which methods lead to the most effective ensembles? Under the heading of modular approaches we begin by considering a divide-and-conquer approach by which a function is automatically decomposed into a number of subfunctions which are treated by specialist modules. Other modular approaches are also identified and considered, for whilst the divide-and-conquer approach is designed to improve performance, the term modularity can be given a wider interpretation. The broadly defined topic of modularity includes the explicit decomposition of a task based on the designer's understanding, and the exploitation of specialist modules in order to accomplish tasks which could not be performed by a monolithic net.

1 Introduction

In the earlier halcyon days of neural computing, it seemed possible to accomplish significant feats through the use of monolithic nets. Indeed, sometimes it was the fact that a unitary mechanism was being used that was of most interest (eg Rumelhart and McClelland's past tense model, Rumelhart and

McClelland, 1986). More recently, it has become apparent that there are many tasks which cannot be effectively solved by means of training a simple unitary net.

There are a number of practical advantages to either decomposing a task into subtasks, or combining several different solutions to the same task; the most significant one for the present purpose being that of improved performance. Decomposition into subtasks can lead to a reduction of training times, it can make an overall system easier to understand and to modify, it may result in a solution to a task which would not have been achieved through the use of a single net, or it can result in better performance than that achieved without decomposition. There may also be theoretical reasons for taking a modular approach. Clearly in terms of cognitive modelling, and computational neuroscience we need to think in terms of modules, and the way in which they can be combined. In addition to the continued interest in modularity, much interest has recently been expressed in improving the reliability and accuracy of neural net generalisation through combining several nets trained on the same task. All of these examples are instances of the combining of nets and fall under the combined remit of this Special Issue, and its companion (*Connection Science*, **9**, 1). An important point is that it is possible to formulate the problem of combining nets at an abstract level, such that it is relevant to both an applications-driven approach, to biological/cognitive questions, and to theoretical analysis. It should be noted too that, although the focus of this Special Issue is on combining Artificial Neural Nets, many of the techniques discussed here are applicable to a much wider variety of statistical methods.

It is possible to identify two main approaches to combining neural networks, each of which is represented in a Special Issue. First, there is the *ensemble-based* approach (sometimes termed the committee framework) by which a set of nets is trained on what is essentially the same task, and then the outputs of the nets are combined (the present Special Issue, *Combining Artificial Neural Nets: Ensemble Approaches*). The aim is to obtain a more reliable and accurate ensemble output than would be obtained by selecting the best net. This can be contrasted with a *modular* approach (companion Special Issue, *Combining Artificial Neural Nets: Modular Approaches, Connection Science*, **9**, 1). Under a narrow definition of modularity, a problem is decomposed into a number of subtasks. Such decomposition may be accomplished either by explicit means, (explicit decomposition), or automatically (automatic decomposition). A wider definition of modular neural nets is that each net should be self-contained or autonomous (see also Fodor's account of informationally encapsulated modules, Fodor, 1983). The wider definition includes in its scope hybrid systems and specialist systems in which the exploitation of the specialist capabilities of different modules makes it possible to solve tasks which could not otherwise have been solved by a single net.

The aim of the present review is to provide an accessible introduction to the combining of artificial neural networks, so as to set the papers in the Special Issues in context. There is of course overlap between this review and the other papers in this issue, but our aim is to provide an overview of the main themes. Admittedly this overview is partisan, and reflects the author's assessment of the field and its interrelationships. Nonetheless, hopefully others will recognise the themes identified here as important, even if they are aware of omissions. In the sections that follow, I shall begin by considering ensemble-based approaches; briefly outlining some of the methods by which nets can be created for use in ensembles, and methods for combining a set of nets to form an ensemble, and following this with a discussion of the likely effectiveness of different methods. This discussion is followed by a consideration of some of the main issues underlying modular approaches. Outlines of the papers in this Special Issue are incorporated into the review as appropriate.

2 Ensemble-based approaches

The defining characteristic of an ensemble-based approach is that it involves combining a set of nets each of which essentially accomplishes the same task. The use of an ensemble can provide an effective alternative to the tradition of generating a population of nets (for example by using different random initializations of the weights), and then choosing the one with the best performance, whilst discarding the rest. The basic idea underlying the ensemble-based approach is to find ways of exploiting instead of ignoring the information contained in these redundant nets.

Combining estimators to improve performance has quite a long history, although it has recently received more attention in the neural net community. Research on combining estimators in neural computing can be traced back to Nilsson (1965), and is found in a number of fields such as econometrics (forecast combining, Clemen, 1989; Granger, 1989), machine learning (evidence combination, Barnett, 1981) and software engineering (diversity, Littlewood and Miller, 1989; Knight and Leveson, 1986). In terms of neural computing, there are two main issues; first the creation, or selection, of a set of nets to be combined in an ensemble; and second, the methods by which the outputs of the members of the ensemble are combined. I shall outline the main methods of creation, and combination in turn, before considering the available guidelines about which methods should be adopted under which circumstances.

2.1 Methods for creating ensemble members

Clearly there is no advantage to combining a set of nets which are identical; identical that is, in that they *generalise in the same way*. The emphasis here is on the similarity or otherwise of the pattern of generalisation. In principle, a set of nets could vary in terms of their weights, the time they took to converge, and even their architecture (eg the number of hidden units) and yet constitute essentially the same *solution*, since they resulted in the same pattern of errors when tested on a test set. There are a number of parameters which can be manipulated in efforts to obtain a set of nets which generalise differently. These include the following: initial conditions, the training data, the typology of the nets, and the training algorithm. We can provide an overview of the main methods which have been employed for the creation of ensemble members, whilst providing more information about methods which involve varying the data, since that is the approach which has most commonly been taken.

- Varying the set of initial random weights: A set of nets can be created by varying the initial random weights from which each net is trained whilst holding the training data constant.

- Varing the topology: A set of nets can be created by varying the topology or architecture, and training with a varying number of hidden units whilst holding the training data constant.

- Varying the algorithm employed: The algorithm used to train the nets could be varied whilst holding the data constant. Our concern here is with the use of Artificial Neural Nets, but the members of an ensemble could be created using a variety of statistical techniques.

- Varying the data: The methods which seem to be most frequently used for the creation of ensembles are those which involve altering the training data. There are a number of different ways in which this can be done which include: sampling data, disjoint training sets, boosting and adaptive resampling, different data sources, and preprocessing.These are considered individually below, although it should be noted that ensembles could be created using a combination of two or more of these techniques (e.g. sampling *plus* preprocessing, see Raviv and Intrator, this issue).

Sampling data: A common approach to the creation of a set of nets for an ensemble is to use some form of sampling technique, such that each net in the ensemble is trained on a different subsample of the training data. Resampling methods which have been used for this purpose include cross-validation (Krogh and Vedelsby, 1995), and bootstrapping (Breiman, 1996b), although in statistics the methods are better known as techniques for estimating the error of a predictor from limited sets of data. For example, in bagging (Breiman, 1996b) a training set containing N cases is perturbed by sampling with replacement (bootstrap) N times from the training set. The perturbed data set may contain repeats. This procedure can be repeated several times to create a number of different, although overlapping, data sets. Such statistical resampling techniques are particularly useful where there is a shortage of data.

Disjoint training sets: A similar method to the above is the use of disjoint, or mutually exclusive training sets, i.e. sampling without replacement (e.g. Sharkey, Sharkey and Chandroth, 1995a,b). There is then no overlap between the data used to train different nets.

Boosting and Adaptive resampling: Schapire (1990) showed that a series of weak learners could be converted to a strong learner as a result of training the members of an ensemble on patterns that have been filtered by previously trained members of the ensemble. A number of empirical studies (eg Drucker et al., 1994) support the efficacy of the boosting algorithm, although a problem with this method is that it requires large amounts of data. Freund and Schapire (1996) have proposed an algorithm developed in the context of boosting. Essentially the basis of this algorithm is that training sets are adaptively resampled, such that the weights in the resampling are increased for those cases which are most often missclassified. Breiman (1996a) explores some of the differences between the Freund and Schapire algorithm; contrasting its effectiveness to that of bagging, and concluding, on the basis of empirical and analytic evidence, that Freund and Schapire's algorithm is more successful than bagging at variance reduction.

Different data sources: another method of varying the data on which nets are trained is to use data from different input sources. This is possible under circumstances in which, for instance, more than one sensor is used, and it is particularly applicable where the sensors are designed to pick up different kinds of information. For example, picking up fuel injection faults in a diesel engine using either a measure of engine cylinder pressure, or engine cylinder temperature (Sharkey, Sharkey and Chandroth, 1995a,b).

Preprocessing: the data on which nets are trained can also be varied by using different preprocessing methods. For example, different feature sets can be extracted from the raw data. Alternatively, the input data for a set of nets could be distorted in different ways; for example by using different pruning methods (see Tumer and Ghosh, this issue), by injecting noise (see Raviv and Intrator, this issue), or by using non-linear transformations (Sharkey and Sharkey, 1995a; Sharkey, Sharkey and Chandroth 1995a,b).

2.2 Methods of combining

Once a set of nets has been created, an effective way of combining their several outputs must be found. There are several different methods of combining, and since a number of reviews of the topic already exist, (e.g. Jacobs, 1995; Genest and Zideck, 1986; Xu et al., 1992), I shall do no more than briefly outline some of the more common methods. Zhilkin and Somorjai, (this issue) also provide a review of combining methods.

Averaging and weighted averaging: Linear opinion pools are one of the most popular aggregation methods, and refer to the linear combination of the outputs of the ensemble members' distributions with the constraint that the resulting combination is itself a distribution (see Jacobs, 1995). An single output can be created from a set of net outputs via simple averaging, (e.g. Perrone and Cooper, 1993), or by taking a weighted average (e.g. Perrone and Cooper, 1993; Hashem and Schmeiser, 1993).

Non-linear combining methods: Other non-linear combining methods that have been proposed include Dempster-Shafer belief-based methods, (e.g. Rogova, 1994), combining using rank-based information (e.g. Al-Ghoneim and Kumar, 1995), voting (e.g. Hansen and Salamon, 1990), and order statistics (Tumer and Ghosh, 1995).

Supra Bayesian: Jacobs (1995) contrasts supra Bayesian with linear combinations. The underlying philosophy of supra Bayesian approach is that the opinions of the experts are themselves data. Therefore the probability distribution of the experts can be combined with its own prior distribution.

Stacked generalisation: Under stacked generalisation (Wolpert, 1992) a non-linear net learns how to combine the networks with weights that vary over the feature space. The outputs from a set of level 0 generalisers are used as the input to a level 1 generaliser, which is trained to produce the appropriate output. The term 'stacked generalisation' is used by Wolpert (1992) to refer both to this method of stacking classifiers, and also to the method of creating a set of ensemble members by training on different partitions of the data. It is also possible to view other methods of combining, such as averaging, as instances of stacking with a simple level 1 generaliser. The same idea has been adapted to regression tasks, where it is termed 'stacked regression', (Breiman, 1993). A comprehensive exploration of stacking is reported by LeBlanc and Tibshirani, (1993).

2.3 Choosing a combining method

In considering which methods should be adopted for creating and combining the members of an ensemble, it is helpful to consider the likely effect that combining nets in an ensemble will have. Such a consideration is likely to rely on the concepts of bias and variance. Much has been made recently of the fact that the error of a predictor can be expressed in terms of the *bias* squared plus the *variance* (see Geman et al., 1992 for a detailed presentation of these concepts, and for further discussion, see Bishop, 1995; Parmanto, Munro and Doyle, this issue; Raviv and Intrator, this issue; Rosen, this issue). A net can be trained to construct a function $f(\mathbf{x})$, based on a training set $(x_1, y_1), ..., (x_n, y_n)$ for the purpose of approximating y for previously unseen observations of x. Following Geman et al. (1992) we shall indicate the dependence of the predictor f on the training data by writing $f(\mathbf{x}; D)$ instead of $f(\mathbf{x})$. Then the mean squared error of f as a predictor of y may be written

$$E_{\mathcal{D}}[(f(\mathbf{x};\mathcal{D}) - E[y|\mathbf{x}])^2]$$

where $E_{\mathcal{D}}$ is the expectation operator with respect to the training set $\mathcal{D}$, (i.e. the average of the set of possible training sets), and $E[y|\mathbf{x}]$ is the target function. Now the bias/variance decomposition gives us,

$$E_{\mathcal{D}}[(f(\mathbf{x};\mathcal{D}) - E[y|\mathbf{x}])^2] \quad =$$

$$
\begin{aligned}
&(E_{\mathcal{D}}[f(\mathbf{x};\mathcal{D})] - E[y|\mathbf{x}])^2 \quad \text{"bias"} \\
&+E_{\mathcal{D}}[(f(\mathbf{x};\mathcal{D}) - E_{\mathcal{D}}[f(\mathbf{x};\mathcal{D})])^2] \quad \text{"variance"}
\end{aligned}
$$

The bias and variance of a predictor can be estimated when the predictor is trained on different sets of data sampled randomly from the entire possible set. The bias of a net can be intuitively characterised as a measure of its ability to generalise correctly to a test set once trained (reflecting the average output over the set of possible training sets). The variance of a net can be similarly characterised as a measure of the extent to which the output of a net is sensitive to the data on which it was trained, i.e. the extent to which the same results would have been obtained if a different set of training data were used.

There is a tradeoff between bias and variance in terms of training nets; the best generalization requires a compromise between the conflicting requirements of small variance and small bias. It is a tradeoff between fitting the training data too closely (high variance), and taking no notice of it all (high bias). What is required of a net that is to generalise well following training on noisy or unrepresentative data[1] is to take sufficient account of the data, but to avoid overfitting (low variance, low bias).

The bias and variance can be approximated by an average over a fixed number of possible training sets. Krogh and Vedelsby (1995) provide an account of the bias and variance in an ensemble, expressing the bias-variance relation in terms of an ensemble average, instead of an average over possible training sets (which means that the ensemble members could be created by a variety of methods, see Section 2.1, as well as by varying the training set). Krogh and Vedelsby's account is made use of by Opitz and Shavlik, this issue. In terms of an ensemble of nets, the bias measures the extent to which the ensemble output averaged over all the ensemble members differs from the target function, whilst the variance is a measure of the extent to which the ensemble members disagree (Krogh and Vedelsby use the term 'ambiguity' to refer to this disagreement).

An ensemble which exhibits high variance should also show a low correlation of errors. It has simultaneously become evident to many neural net researchers (and it is a theme that is well represented in this issue), that the main determinant of the effectiveness of an ensemble is the extent to which the members are 'error-independent' (Rogova, 1994), in the sense

[1]If the data were not noisy, and were sufficiently representative of the test set to permit good generalisation, then there would be no problem with overfitting.

that they make different errors (or to put it another way, show different patterns of generalisation). Wolpert (1992) for instance points out that '..the more each generalizer has to say (which isn't duplicated in what other generalizers have to say), the better the resultant stacked generalization..' And for Jacobs (1995), '..The major difficulty with combining expert opinions is that these opinions tend to be correlated or dependent..' This same point is made in other areas, such as software engineering (Eckhardt and Lee, 1985; Littlewood and Miller, 1986), and forecasting (Guerard Jr and Clemen, 1989). The ideal, in terms of ensembles of artificial neural nets, would be a set of nets which did not show any coincident errors. That is, each of the nets generalised well (low bias component of error), and when they did make errors on the test set, these errors were not shared with any other nets (high variance component of error).

Rather than just considering the relative contribution of bias and variance to the total error, or measuring the error correlation, it is also possible to distinguish different types of error patterns that an ensemble may exhibit when tested. Sharkey and Sharkey (1995a) present an account of four different types of error pattern which may be exhibited by an ensemble with respect to a validation test set, (although they use the term 'diversity'). These range from Type 1 to Type 4. In Type 1 Diversity, there are no coincident errors, and when errors occur on one net they are not shared with any other ensemble member. In Type 2 Diversity there are coincident errors, but the majority is always correct. In Type 3 Diversity, the majority is not always correct, but the correct output is always produced by at least one net. In Type 4 Diversity, the majority is not always correct, and there are some inputs which fail on all the ensemble members, but there is some difference between the errors made by different nets, and therefore some advantage to be gained from combining. A major advantage of this typology is that it makes it possible to quantify the level of error independence achieved by an ensemble.

Once the importance of the error correlation between the nets has been recognised, the main approaches which can be adopted are are:

1. To take account of the dependency between the ensemble members when choosing a method of combining.

2. To select and/or create nets that are relatively independent.

Taking account of the dependency between nets: The extent to which the outputs of a set of nets are correlated gives a strong indication about how they should be combined. For example, if on a classification problem, an ensemble does not exhibit any coincident failures with respect to a validation set (Type 1 Diversity), then combining the nets by means of a simple majority vote will produce good results. Good results will also be obtained if a simple majority vote is used to combine nets which do share coincident errors, but where the majority is always correct (Type 2 Diversity). Where there are overlapping errors, more complex methods of combination, such as stacked generalisation are likely to be appropriate. Of course, another way of taking account of the dependency between nets is to select nets for effective combination on the basis of an analysis of the dependency among their outputs (see next section, and Hashem, this issue).

Creation and Selection of nets for effective combination: A more recent trend, as evidenced in this issue, is to actively select, from a larger pool, a set of nets which can be combined effectively, instead of comparing different methods of combining. A discussion of the selection of nets for effective combination can be found in Perrone and Cooper (1993), who suggest not including in an ensemble near duplicate nets which exhibit a high degree of correlation.

The idea of selecting nets for effective combination is implied by the conclusion that linear combining methods such as weighted averaging suffer when the outputs of the ensemble members are correlated (see Tumer and Ghosh, this issue; Hashem, this issue). There are two papers in this issue which particularly focus on the *selection* of nets for effective combination; Hashem, (this issue), and Opitz and Shavlik (this issue).

Hashem(this issue) details the harmful effects that collinearity or linear dependence among the members of an ensemble may have on the estimation of the optimal weights for combining. He discusses the idea of selecting nets for combining, and presents an approach whereby the selection of candidate nets for combining via weighted averaging is guided by diagnostics of collinearity between potential members. His experimental results demonstrate improved ensemble results as a result of such selection. Better results are obtained than were produced by two alternative methods; selecting the best network, and taking the simple average of all the candidates without selection.

Opitz and Shavlik (this issue) present an algorithm that uses genetic algorithms to actively search for ensemble members which generalise well, but which disagree as much as possible. The standard genetic operators, crossover and mutation, are used to create new individuals from an initial set. The most fit members (in terms of generalisation and disagreement, or diversity) then form the next generation, and the process is repeated until 'a stopping criterion is reached'. Once found, the ensemble members are combined using weighted averaging. Opitz and Shavlik report experiments on four real-world domains which indicate that their method can, under some circumstances, outperform some existing ensemble approaches (choosing the best network, and bagging (Breiman, 1996b)). The algorithm can incorporate prior knowledge in order to create a more accurate ensemble.

The selection of nets for effective combination usually relies on the generation of a pool of nets through the application of one of the methods for creating ensemble members (see Section 2.1). Some papers in this issue focus on the presentation of new methods for creating candidate members for ensembles (eg Raviv and Intrator, this issue; Rosen, this issue). Others have chosen to conduct an empirical comparison of the relative effectiveness of different methods for creating candidate ensemble members, (Tumer and Ghosh, this issue; Parmanto, Munro and Doyle, this issue) or of the relative effectiveness of different combining methods (Zhilkin and Somorjai, this issue).

Raviv and Intrator (this issue) present a method for the creation of ensemble members that involves a combination of bootstrap sampling of data, the addition of variable amounts of noise to the inputs, and weight decay. A number of different noise levels are assessed with reference to the ensemble performance. The ensemble members are then combined by means of a simple average. Raviv and Intrator apply their method of noisy bootstrap + weight decay to (a) the two-spiral problem, a highly non-linear noise free dataset; and (b) a highly linear data set, the Cleveland Heart Data. The improved ensemble performance is discussed with reference to the role of variance in ensemble performance.

Rosen (this issue) takes a different but related approach to the above in that he is concerned with reducing the error correlation between ensemble members. He presents a decorrelation network training method in which the members of an ensemble are trained not only to produce a desired output but also to have their errors be linearly decorrelated with other networks. This is accomplished through the addition of a correlation penalty term, such that nets attempt both to minimize the error between the target and output, and to decorrelate their errors with those from previously trained networks. A comparison between decorrelation ensemble networks (trained to be decorrelated), and regular ensemble networks on three tasks, indicates that performance was improved when decorrelation network training was used. Rosen suggests the method is particularly applicable when the data is too limited to permit the creation of disjoint subsets.

Tumer and Ghosh (this issue) argue that the extent to which an ensemble results in improved performance, is more a factor of the members contained in an ensemble, than of the combining method used. The need to reduce the correlation among the members of an ensemble can be quantified in Bayesian terms (Tumer and Ghosh, 1995, Tumer and Ghosh, 1996). In this paper, Tumer and Ghosh present an empirical comparison of four methods for reducing correlations, (cross-validation, pruning the inputs, resampling, and data partitioning according to spatial similarity) all of which involve varying the data set. The reduction of correlation between ensemble members improved the performance of an ensemble, but its effectiveness was mitigated by the consequent reduction in the size of the training set. The authors conclude that it is important to reduce the correlations without increasing the error rates (i.e. to find methods for reducing the variance, without increasing the bias).

Parmanto, Munro and Doyle (this issue) examine the effect of a variety of ensemble creation methods (varying initial conditions, cross validation and bootstrap) in terms of the resulting decomposition of the error into variance and bias. Extensive simulation results are presented showing the effect of applying the different ensemble creation methods to data sets with different levels of noise, and to medical diagnosis sets. Better ensemble results are obtained when resampling techniques are used to vary the training data, than when ensemble members are created by varying the initial conditions. Ensemble methods provide an effective means of reducing the variance, and therefore of overcoming the problem of overfitting to the training data. Improvement due to ensemble averaging is greatest when the data is noisy and the training set is small. However, like Tumer and Ghosh (this issue), Parmanto et al. note that smaller training sets can result in decreased performance where the bias component of the error is high.

Zhilkin and Somorjai (this issue) report an empirical comparison of the effectiveness of different methods of combining classifiers of Magnetic Resonance spectra - evaluating their relative ability to improve classification performance beyond that achieved by a single classifier. They provide a brief review of combining methods, and go on to consider logistic regression, linear combination, entropy and confidence factor approaches, a fuzzy integral approach, and a stacked generalisation scheme. These methods are applied to both artificial and real MR spectra data. The authors conclude that the effectiveness of different aggregation methods depends on both the data, and on the preprocessing techniques used.

Wolpert (Wolpert, 1992) has described the available guidance on the choice of methods for generating ensemble members (or level 0 generalizers in his terms), as a 'black art'. However, it would seem that a consensus is beginning to emerge from these papers, and others in the field. The most obvious point is that ensemble methods can provide an effective means of improving performance. This performance improvement is usually the result of a reduction in variance, rather than bias, since the effect of ensemble averaging is to reduce the variance of a set of nets.[2] Therefore, an effective approach is to create and/or select a set of nets that exhibits high variance, but low bias, since the variance component can be removed by combining. There is also some agreement about which external parameters can be manipulated effectively in order to accomplish this; namely those which involve altering the data.

Varying the data on which a set of nets are trained is more likely, it appears, to result in a set of nets that can be combined effectively than varying for instance the set of initial conditions from which they are trained, or their topology. The conclusion about the relative ineffectiveness of varying the initial conditions is supported by the results of Parmanto, Munro and Doyle (this issue), and those of Sharkey, Neary and Sharkey (1995). Although the point has been made that backpropagation is sensitive to initial conditions (Kolen and Pollack 1990) the available evidence suggests that although variations in initial conditions may affect the speed of convergence, or whether or not a net converges, the resulting differences in generalisation are likely to be slight. It seems that unless the neural net being trained is low in complexity, often only one function that is compatible with a set of data is found. Therefore, regardless of the initial set of weights, the algorithm, or the topology of the nets, a net that has learned a particular set of data is likely to show the same pattern of generalisation. Of course, it is difficult to argue conclusively against the possiblity that altering the initial conditions of a net could result in significant changes in the pattern of generalisation, but the evidence suggests varying the initial conditions is likely to be less effective than training nets on different data sets.

[2] Although some forms of stacking, i.e. the use of a level 1 generaliser, may reduce bias, (eg Kim and Bartlett, 1995).

Methods which involve varying the data include methods of sampling (eg bootstrap, or cross-validation); and the use of different methods of preprocessing, or distortion of the inputs (Raviv and Intrator, for instance inject noise, whilst Tumer and Ghosh prune the inputs in different ways). However, any of these methods are best combined with an approach which emphasises the testing and selection of ensemble members, for it cannot be assumed that adopting a particular approach ensures that error independence will be achieved. We shall expand upon this point with respect to the notion of training set representativeness, and argue that it cannot be assumed even that using disjoint training sets will lead to a set of nets that show a low number of coincident errors.

Our argument here is that *disjoint training sets will not necessarily result in low error correlations.* This point can be explained with reference to the concept of training set *representativeness* (see Denker et al. 1987 and Sharkey and Sharkey, 1995b for further discussion of the notion of training set representativeness). A representative training set is one which leads to a function being inferred which is similar to that which generated the test set. A representative training set will therefore lead to good generalisation. The problem is however, that two representative training sets, even if the data that defined them did not overlap at all, could still lead to very similar functions being inferred, with the result that their pattern of errors on the test set will be very similar. For instance, think of a simple classification determined by a boundary (i.e. a square wave boundary) where the output is 1 on one side of the boundary, and 0 on the other. There is a very large, or unbounded number of different combinations of data points which could be chosen as boundary conditions, but which would yield the same, or nearly the same pattern of generalisation. In the same way, the data points which make up a training set should not overlap with those in a test set, but it is to be hoped that they result in almost the same function being inferred.

On the other hand, if a candidate set of nets were trained using unrepresentative training sets, the resulting generalisation performance would be poor. The nets might each infer quite different functions, and show different patterns of generalisation to the test set, but as the amount of errors increases so does the probability that the errors that they make on the test set will overlap. There is therefore a delicate balance between training set representativeness and error correlation. What is needed is several training sets, all of which are representative and lead to good generalisation, but which exhibit a minimum number of coincident failures. The extent to which they exhibit coincident failures (or the determination of the type of diversity they exemplify) can only be determined through a process of testing the performance of selected ensembles.

There is an alternative to the ensemble approach we have been discussing, where the members of an ensemble are all trained as general function approximators, and that is to decompose a task into a number of subtasks. This is likely to remove the problem of correlated errors, for it results in modules that know more about certain aspects of the data. However, now it makes more sense to look at the generalisation performance of the combined system than to look at the generalisation performance of the component experts on the entire range of the data.[3]

3 Modular approaches

In terms of combining nets, the topic of modularity could be considered to be wider than that of ensembles, since the notion of what is a module can be given a very broad interpretation. For instance, hybrid systems where different architectures are joined together to accomplish disparate tasks can also be considered examples of a modular approach. As discussed in the introduction, a narrow definition of modularity could restrict it to instances in which a task is decomposed into subtasks, whilst a wider definition could be extended to modules were considered to be autonomous or informationally encapsulated (such that their internal computation was unaffected by other modules, and the only means of inter-modular communication was in terms of their inputs and outputs). The topic of modularity is less constrained, and consequently harder to define, than an ensemble-based approach, but it is nonetheless possible to identify some of the themes and issues which characterise research on modularity in general. These themes are represented in the companion Special Issue to the present one; Special Issue on Combining Artificial Neural Nets: Modular Approaches, (*Connection Science*, **9**, 1).

First we can consider the reasons for adopting a modular approach. A modular approach can be adopted for the purpose of improving performance: where a task could be accomplished with a monolithic net, but better performance is achieved if it is broken down into a number of modules. Alternatively, it might not be possible to accomplish the task in question unless the specialist capabilities of a number of modules were exploited,

[3]Although an examination of the generalisation performance of modular experts can provide a vehicle for comparing and contrasting the ensemble and modular approach (Jacobs, 1996).

and therefore modularity is required in order to extend the capabilities of a single net. Another reason for modularity that is sometimes given is that it represents an approach which is more coherent in terms of biology/cognition/neurophysiology - there are often clear justifications for particular subdivisions when the aim is to model aspects of brain functions. And a final reason for taking adopting a modular approach is that it can lead to a simpler more coherent system, that is easier to understand and to modify, and which results in shorter training times than training a single net.

An important question in modular approaches is that of how the modules are combined, or communicate in order to accomplish the task in question. Another issue underlying the modular approach is how the task is decomposed into modules. Decomposition may be accomplished automatically, or explicitly. Where the decomposition into modules is explicit, this usually relies on a strong understanding of the problem. The division into subtasks is known prior to training (eg Hampshire and Waibel, 1989), and improved learning and performance can result (eg Waibel et al., 1989). An alternative approach is one in which *automatic decomposition* of the task is undertaken, characterised by the blind application of a data partitioning technique. Automatic decomposition is more likely to be carried out with a view to improving performance, whilst explicit decomposition might either have the aim of improving performance, or that of accomplishing tasks which either could not be accomplished using a monolithic net, or could not be accomplished either as easily, or as naturally.

Automatic decomposition of a task for the purposes of improved performance is an approach which is closely related to the ensemble-based one we have already considered. Under the *divide and conquer* approach of Jacobs and Jordan (Jacobs et al., 1991; Jordan and Jacobs, 1994; Peng, Jacobs and Tanner, 1995) complex problems are automatically decomposed into a set of simpler problems. Mixtures-of-experts (Jacobs et al., 1991) and Hierarchical mixtures-of-experts (Jordan and Jacobs, 1994) partition the data into regions and fit simple surfaces to the data that fall in each region. Expert nets learn to specialize onto subtasks and to cooperate by means of a gating net. The regions have 'soft' boundaries, which means that data points may lie simultaneously in multiple regions. The mixtures-of-experts model consists of a number of expert networks, combined by means of a gating network which identifies the expert, or blend of experts, most likely to approximate the desired response. The hierarchical extension of the mixtures-of-experts model is a tree-structured model which recursively divides each region into sub-regions. Such decomposition ensures that the errors made by the expert nets will not be correlated, for they each deal with different data points.

There are similarities between the mixtures-of-experts approach, and an ensemble-based one; the underlying aim of both is the improvement of performance, and both can involve linear combinations of their components. However, the approaches are distinct, in that the mixture-of-experts approach assumes that each data point is assigned to only one expert (mutual exclusivity) whereas ensemble combination makes no such assumption, and each data point is likely to be dealt with by all the component nets in an ensemble. In electronic discussions, Jordan has suggested that mixtures of experts are best thought of as another kind of statistical model, such as hidden Markov models. Thus, one of the members of an ensemble could be a mixtures-of-experts approach to a particular task, whilst other members were trained on the task using other techniques.

The mixtures-of-experts approach is significant in the field, (and one of the correlation-reduction methods used by Tumer and Ghosh, this issue, is based on it) but there are other important modular approaches. The modular approaches represented in the companion Special Issue, Combining Artificial Neural Nets: Modular Approaches (*Connection Science*, **9**, 1) include examples both of tasks that have been decomposed (either automatically, or explicitly) in order achieve improved performance, and of problems which require a modular approach and could not have been solved achieved through the use of a unitary net.

4 Conclusions

In this paper some of the main ways in which artificial neural nets may be combined in order to improve their performance have been examined, and the papers contained in this issue have been outlined. One of the advantages that can be gleaned from a Special Issue that contains several papers on one topic is that of consolidation, if similar results and conclusions are drawn in a variety of papers. An example of a recurrent theme in this issue is that of the role of error correlation in determining the effectiveness of a combination of nets, although this theme is approached differently in the various papers. As discussed at the end of the section on ensemble-based approaches, a consensus is emerging about the methods by which an effective ensemble can be created. It seems accepted that nets to be combined should share a minimum number of coincident errors; something that can be achieved in different ways in either the mixtures-of-experts, and other modular approaches, or in an ensemble-based approach.

Another advantage which can accrue to a Special Issue is that it should promote a cross-fertilisation of ideas. First, those not in the area can speedily become aware of the main issues and concerns in the area through reading papers which represent some of the dominant concerns in the area. And second, researchers in the area can become aware of closely related approaches which it might be possible to modify and incorporate in their own work. A potential cross-link apparent to the present author for example is the idea of using types of diversity (Sharkey and Sharkey 1995a) in combination with methods for selecting and creating ensemble members. The goal of selecting nets for combination in an ensemble which showed no coincident errors on a validation set (Type 1 Diversity) could be used as a stopping criterion, such that candidate ensemble members would continue to be generated using a creation method (eg noisy bootstrap, Raviv and Intrator this issue), until it was possible to assemble a committee of nets which did not exhibit any coincident errors on a test set. There are of course many other potential crosslinks. It would for instance be interesting to know how a combination of some of the approaches presented here would fare; for example, if the selection of nets using diagnostics of collinearity (Hashem, this issue) were combined with methods for generating nets which disagree. Or if one of the members of an ensemble were created using a mixture of experts approach whilst others used other methods such as standard multi-layer perceptron training.

In addition to the consolidation and cross-fertilisation that might be engendered by a family of related research papers, it is also tempting to speculate where the field might move next. As always, there are unresolved questions which require further work. An example is that of the relationship between the modular automatic decomposition approach (e.g. mixtures-of-experts) and the ensemble-based approach. As Jordan and Jacobs (1995) point out, we still need 'to characterize those classes of problems for which the different approaches are most appropriate.' It also seems clear that an important future direction for research on combining nets must be to begin to move these approaches out into the world of real applications. Reliability and accuracy of neural net performance is particularly important in the domain of real applications, and it would make sense to see the standard adoption of techniques for improving performance by combining solutions.

5 Acknowledgements

Comments by Cesare Furlanello, Joydeep Ghosh, Sherif Hashem, Nathan Intrator, Robert Jacobs, Michael Perrone, and David Wolpert have helped the author to improve this paper. The research on which this paper is based was supported by EPSRC Grant No GR/K84257.

6 References

Al-Ghoneim, K. and Vijaya Kumar, B.V.K. (1995) Learning ranks with neural networks (Invited paper). In Applications and Science of Artificial Neural Networks, Proceedings of the SPIE, volume 2492, pg 446-464.

Barnett, J.A. (1981) Computational methods for a mathematical theory of evidence. In Proc. of IJCAI, pp 868-875.

Bishop, C. (1995) *Neural Networks for Pattern Recognition.* Oxford, Clarendon Press.

Breiman, L. (1993) Stacked regression. Technical Report, University of California, Berkeley.

Breiman, L. (1996a) Bias, Variance and Arcing Classifiers. Technical Report 460.

Breiman, L. (1996b) Bagging predictors, in press, Machine Learning.

Clemen, R. (1989) Combining forecasts: A review and annotated bibliography. *International Journal of Forecasting*, 5, 559-583.

Denker, J., Schwartz, D., Wittner, B., Solla, S. Howard, R., Jackel, L. and Hopfield, J. (1987) Large automatic learning, rule extraction and generalisation. *Complex Systems*, 1, 877-922.

Drucker, H., Cortes, C., Jackel, L.D., LeCun, Y., and Vapnik, V. (1994) Boosting and other ensemble methods. *Neural Computation*, 6(6): 1289-1301.

Eckhardt, D.E. and Lee, L.D. (1985) A theoretical basis for the analysis of multiversion software subject to coincident errors. IEEE Transactions on Software Engineering, Vol SE-11, 12.

Fodor, J.A. (1983) *The Modularity of Mind: An essay on Faculty Psychology*, London, England: A Bradford Book, MIT Press.

Freund, Y. and Schapire, R. (1996) Experiments with a new boosting algorithm, to appear 'Machine Learning: Proceedings of the Thirteenth International Conference', July 1996.

Genest, C., and Zidek, J.V. (1986) Combining probability distributions: A critique and annotated bibliography. *Statistical Science*, 1, 114-148.

Granger, C.W.J (1989) Combining forecasts - twenty years later. *Journal of Forecasting* 8 (3) 167-173.

Gueard Jr., J.B. and Clemen, R.T. (1989) Collinearity and the use of latent root regression for combining GNP forecasts. Journal of Forecasting, 8, 231-238.

Hampshire, J., and Waibel, A. (1989) The meta-pi network: Building distributed knowledge systems for robust pattern recognition. Tech. Rep. CMU-CS-89-166, Carnegie Mellon University, Pittsburgh, PA.

Hansen, L.K. and Salamon, P. (1990) Neural network ensembles. IEEE Transactions on Pattern Analysis and Machine Intelligence, 12(10): 993-1000.

Hashem, S., and Schmeiser, B.(1993) Approximating a Function and its Derivatives using MSE-Optimal Linear Combinations of Trained Feedforward Neural Networks. In *Proceedings of the World Congress on Neural Networks* vol 1, pp 617-620, Lawrence Erlbaum Associates, New Jersey.

Jacobs, R.A. (1995) Methods for combining experts' probability assessments. *Neural Computation*, 7, 867-888.

Jacobs, R.A. (1996) Bias/Variance Analyses of Mixtures-of-Experts Architectures. To appear in *Neural Computation.*

Jacobs, R.A., Jordan, M.I., Nowlan, S.J. & Hinton, G.E. (1991) Adaptive mixtures of local experts. *Neural Computation*, 3, 79-97.

Jordan, M.I. and Jacobs, R.A. (1994) Hierarchical mixtures of experts and the EM algorithm. *Neural Computation*, 6, 181-214.

Jordan, M.I. and Jacobs, R.A. (1995) Modular and Hierarchical Learning Systems. In M.A. Arbib (Ed) *The Handbook of Brain Theory and Neural Networks*, pp 579-581.

Kim, K. and Bartlett, E.B. (1995) Error Estimation by Series Association for Neural Network Systems. *Neural Computation*, 7, 799-808.

Krogh, A. and Vedelsby, J. (1995) Neural network ensembles, cross validation and active learning. In Tesauro, G., Touretzky, D.S. and Leen, T.K. (Eds) *Advances in Neural Information Processing Systems 7*, MIT Press.

Knight, J.C. & Leveson, N.G. (1986) An experimental evaluation of independence in multiversion programming. *Trans on Software Eng.*, vol SE-12, no 1

LeBlanc, M., and Tibshirani, R. (1993) Combining estimates in regression and classification. Paper available from ftp site: utstat.toronto.edu.

Littlewood, B., & Miller, D.R. (1989) Conceptual modelling of coincident failures in multiversion software. IEEE Trans on Software Engineering, 15, 12.

Kolen, J.F. and Pollack, J.B. (1990) Backpropagation is sensitive to initial conditions. TR 90-JK-BPSIC

Granger, C.W.J (1989) Combining forecasts - twenty years later. *Journal of Forecasting* 8 (3) 167-173.

Hampshire, J. and Waibel, A. (1989) The Meta-Pi network: Building distributed knowledge representations for robust pattern recognition. Technical Report CMU-CS-89-166, Pittsburgh, PA: Carnegie-Mellon University.

Nilsson, N.J. (1965) *Learning Machines: Foundations of Trainable Pattern-Classifying Systems* McGraw Hill, NY.

Peng, F., Jacobs, R.A., and Tanner, M.A. (1996) Bayesian Inference in Mixtures-of-Experts and Hierarchical Mixtures-of-Experts Models with an application to Speech Recognition. Accepted for publication in *Journal of the American Statistical Association.*

Perrone, M. and Cooper, L.N. (1993) When networks disagree: Ensemble methods for hybrid neural networks. In R.J. Mammone (Ed) *Neural Networks for Speech and Image Processing*, Chapman Hall.

Rogova, G. (1994) Combining the results of several neural network classifiers. *Neural Networks*, 7(5) 777-781.

Rumelhart, D.E. and McClelland, J.L.,(1986) On learning the past tense of English verbs, In (Eds) D.E. Rumelhart, J.L. McClelland, and PDP Research Group, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, Cambridge, MA: MIT Press.

Sharkey, A.J.C, Sharkey, N.E. and Chandroth, G.O. (1995a) Neural Nets and Diversity. Proceedings of the 14th International Conference on Computer Safety, Reliability and Security, Belgirate, Italy, 11-13 October 1995 pp 375-389

Sharkey, A.J.C., Sharkey, N.E. and Chandroth, G.O. (1995b) Diverse Neural Net solutions to a Fault Diagnosis Problem . Research Report, CS-95-10 Department of Computer Science, University of Sheffield. To appear in *Neural Computing and Applications.*

Sharkey, N.E., Neary, J. and Sharkey, A.J.C. (1995) Searching weight space for backpropagation solution types. In L.F. Niklasson and M.B. Boden (Eds) *Current Trends in Connectionism*, Lawrence Erlbaum Associates, Hillsdale, NJ., pp 103-121

Sharkey, A.J.C. and Sharkey, N.E. (1995a) How to improve the reliability of Artificial Neural Networks. Research Report CS-95-11, Department of Computer Science, University of Sheffield.

Sharkey, N.E. & Sharkey, A.J.C. (1995b) An Analysis of Catastrophic Interference, *Connection Science*, 7, 3/4, 313-341.

Schapire, R.E. (1990) The strength of weak learnability. *Machine Learning*, 5(2), 197-227.

Thria, S., Mejia, C., Badran, F., and Crepon, M. (1992) Multimodular architecture for remote sensing operations, in *Advances in Neural Information Processing Sytsems 4* (J. E. Moody, S.J. Hanson, and R.P. Lippmann, Eds), San Mateio, CA: Morgan Kaufmann, pp 675-682.

Tumer, K. and Ghosh, J. (1995) Theoretical foundations of Linear and Order Statistics Combiners for Neural Pattern Classifiers. TR-95-02-98, The Computer and Vision Research Center, University of Texas, Austin, 1995. (Available from http://www.lans.ece.utexas.edu under select publications – tech reports)

Tumer, K. and Ghosh, J. (1996) Analysis of Decision Boundaries in Linearly Combined Neural Classifiers, *Pattern Recognition*, 29, 2, pp 341-348.

Waibel, A., Sawai, H., and Shikano, K. (1989) Modularity and scaling in large phonemic neural networks. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 37 (12): 1888-1898.

Wolpert, D.H. (1992) Stacked generalization. Neural Networks, 5, 241-259.

Xu, L., Krzyzak, A. and Suen, C.Y. (1992) Methods of combining multiple classifiers and their applications to handwriting recognition. IEEE Trans. Systems, Man, Cybernet., 22, 418-435.