Adaptation of cases for case based forecasting with neural network support

— Source link ↗

Juan M. Corchado, Brian Lees

Related papers:

- Unsupervised neural method for temperature forecasting

- Case-based reasoning: foundational issues, methodological variations, and system approaches

- Mathematical model for dynamic case-based planning

- Ovamah: Multiagent-based adaptive virtual organizations

- Hybrid artificial intelligence methods in oceanographic forecast models

1. Adaptation of cases for case-based forecasting with neural network support

Corchado J. M.
Artificial Intelligence Research Group
Escuela Superior de Ingeniería Informática, University of Vigo,
Campus Universitario As Lagoas, Edificio Politécnico, Ourense, 32004, Spain
Email: corchado@ei.uvigo.es

Lees B.
Applied Computational Intelligence Research Unit
Department of Computing and Information Systems
University of Paisley, Paisley, PA1 2BE, U.K.
Email: lees-ci0@paisley.ac.uk

1.1 Introduction

This chapter describes the application of a hybrid artificial intelligence approach to prediction in the domain of oceanography. A hybrid artificial intelligence strategy for forecasting the thermal structure of the water ahead of a moving vessel is presented. This approach combines the ability of a case-based reasoning system for identifying previously encountered similar situations and the generalising ability of an artificial neural network to guide the adaptation stage of the case-based reasoning mechanism. The system has been successfully tested in real time in the Atlantic Ocean; the results obtained are presented and compared with those derived from other forecasting methods.

Research into artificial intelligence (AI) has produced various hybrid problem-solving methods, which may be applied to give more powerful computer based problem solving capabilities than may be obtained using purely algorithmic methods. The reason for the application of an AI approach is very often precisely because the nature of the problem to be addressed is such that no appropriate algorithm is either known or is applicable. For example, if the knowledge about a problem is incomplete or fuzzy, it may be difficult to select or to develop an algorithm or even an AI approach to solve it. It is in such situations where hybrid AI systems may be effective.

Case-based reasoning systems have proved to be successful in situations where prior experience of solving similar problems is available. But the nature of a complex problem solving situation may be such that there are different aspects of the problem that may best be addressed through the application of several distinct problem solving methodologies. This paper focuses on the combination of case-based reasoning (CBR)

1

and artificial neural networks (ANN) as complementary methods to solve a forecasting problem.

The application of artificial intelligence methods to the problem of describing the ocean environment offers potential advantages over conventional algorithmic data processing methods; an AI approach is, in general, better able to deal with uncertain, incomplete and even inconsistent data. Neural network, case-based and statistical forecasting techniques could be used separately in situations where the characteristics of the system are relatively stable (Lees *et al.*, 1992). However, time series forecasting, based on neural network or statistical analysis, may not provide sufficiently accurate forecasting capability in chaotic areas such as are found near a *front* (i.e. an area where two or more large water masses with different characteristics converge). This paper presents a *universal* forecasting strategy, in which the term universal is taken to mean a forecasting tool which is able to operate effectively in any location, of any ocean.

This chapter shows how a hybrid system can solve the problem of forecasting the surface temperature of the ocean at certain distances ahead of a moving vessel. The case-based reasoning system is used to select a number of stored cases relevant to the current forecasting situation. The neural network retrains itself in real time, using a number of closely matching cases selected by the CBR retrieval mechanism, in order to produce the required forecasted values.

The structure of the chapter is as follows. First the integration of CBR and ANN problem solving methods is introduced; a brief outline of work elsewhere on the integration of CBR and neural network methods is given. The application of a hybrid neural network case-based approach for real-time oceanographic forecasting is presented. Finally, a summary of the experimental results obtained to date are presented, which indicate that the approach performs favourably in comparison with the use of statistical and neural network forecasting methods in isolation.

2. Hybrid Systems

The term *hybrid* refers to systems that consist of one or more integrated subsystems, each of which can have a different representation language and inference technique. The subsystems are assumed to be tied together semantically and influence each other in some way. The goal of hybrid system research includes the development of techniques to increase the efficiency and reasoning power of intelligent systems. For example, some of the work developed with the aim of increasing efficiency makes use of specialised reasoners strategically called by control or supervisor modules that decide which reasoning method to use at different times (Medsker, 1995). Hybrid systems are capable of addressing some practical problems that have been addressed with traditional artificial intelligence approaches. From a fundamental perspective, hybrid systems may also give further insight into cognitive mechanisms and models (Medsker, 1995).

Many researchers are investigating the integration of different AI approaches (Sun, 1996; Lees *et al.*, 1999). The issues under study range from fundamental questions about the nature of cognition and theories of computation to practical problems related to implementation techniques. There are many different directions in this research and several models for integration have been identified.

The three main models of integration of AI techniques that characterise the trends in the research in this area are: the Computational intelligent classification Bezdek (1994), the IRIS Classification Soucek (1991) and the classification of Medsker and Bailey (1992).

1.2.1 The Computational intelligent classification

Bezdek (1994) proposes a framework for thinking about the real goals of research and development in intelligent systems. This model focuses on different levels of integrated activities, systems, and AI technologies. Bezdek defines three levels of increasing complexity, from computational to artificial tasks and then to biological activities. In this model the artificial intelligent components are built on symbolic modules that add relatively small pieces of knowledge to computational processes and data, in order to get closer to biological intelligence. This model does not consider computational and artificial intelligence systems *"intelligent"* by themselves and for Bezdek words such as learning must be very carefully used when applied to low levels of complexity. This is a general classification in which hybrid systems are seen as ways to extend the low level computational intelligence techniques through the artificial intelligence level toward the goal of modelling biological intelligence.

1.2.2 IRIS Classification

Soucek (1991) developed the Integration of Reasoning, Informing and Serving (IRIS) model. This is an architecture that facilitates the combination of software, hardware and system levels present in any intelligent system. The aim of this methodology is to facilitate the design of systems using more efficient technologies, products and services to meet business needs. For Soucek, the development of a hybrid system requires the integration of different scientific disciplines including biology, cognitive psychology, linguistics, epistemology, and computer science.

The IRIS model identifies the need for ten ingredients of integration:

- mixing of technologies (ANN, CBR, Knowledge-based systems, etc.),
- paradigms for integration,
- standard software modules,
- special languages,
- software development tools and environments,
- automated discovery such as interactive intelligent databases and interfaces,
- standard control and automation modules,
- case studies of working applications,

- concurrency - tools for developing and monitoring,
- signal to symbol transformations and pattern to category mappings.

This is not just a classification that describes mechanisms of interaction between AI models. It shows how AI models can be integrated within other computer technologies to create successful knowledge-based systems.

1.2.3 The classification of Medsker and Bailey

Medsker and Bailey (1992) have defined five models of integration from a practical point of view: stand-alone models, transformational, loose coupling, tight coupling and full integration. This classification presents several ways of combining connectionist and symbolic models. The integration between models can be done depending on the problem to be solved and the data and knowledge available.

Stand-Alone model

This model combines intelligent system applications consisting of independent software components. Since the components do not interact in any way, the stand-alone model cannot be considered a real form of integration; it is only used to compare different AI models in order to learn more about them and about the problems to be solved.

Transformational model

This model is similar to the previous one. The difference is that in the transformational model, the system begins as one type and ends up as the other. For example an artificial neural network can be used to identify trends and relationships among data sets and the results obtained with it could be used to develop a knowledge-based system.

Loose coupling model

This is the first true form of integrating artificial intelligent systems. The application is composed of separate intelligent systems that communicate via data files. This model allows the interaction between systems with very different characteristics. Typical cases of this type are:

- **Pre-processors**: In this case an ANN could serve as a front-end that processes data prior to passing it on to a knowledge-based system. Following the principles of this model an ANN can be used to perform data fusion, to remove errors, to identify objects and to recognise patterns. Then the knowledge-based system can play the main role.
- **Post-processors**: In this case for example, a knowledge-based system can produce an output that is passed via a data file to an ANN. The knowledge-based system can perform data preparation and manipulation, classify inputs, etc. and the ANN can then perform functions such as forecasting, data analysis, monitoring, etc.
- **Co-processors**: This type of integration involves data passing in both directions, allowing interacting and co-operative behaviour between the ANN and the knowledge-based system. Although not very often used, this approach has the

potential for solving difficult problems such as incremental data refinement, iterative problem solving and dual decision-making.

- **User interfaces**: An ANN can be used, for example, for pattern recognition to increase the flexibility of user interactions with knowledge-based systems.

Tight coupling model

This model is similar to the previous one; however here the information is passed via memory resident data structures rather than external data files. This improves the interactive capabilities of tightly coupled models in addition to enhancing their performance. The sub-models of this approach are the four mentioned in the previous subsection: pre-processors, post-processors, co-processors and user interfaces. In this type of situation the implementation is more complex and the operation time is smaller than in the previous case.

Fully integrated models

Fully integrated models share data structures and knowledge representations. Communication between different components is accomplished via the dual nature of structures (symbolic and connectionist). Reasoning is accomplished either co-operatively or through a component designated as a controller. Several variations of fully integrated systems exist, for example connectionist knowledge-based systems are one of the most common varieties of this model. They rely on local knowledge representation, as opposed to the distributed representation of most ANN, and reason through spreading activation. Connectionist knowledge-based systems represent relationships between pieces of knowledge, with weighted links between symbolic nodes.

Each of the three classifications here presented considers the hybridisation process from different points of view. The Computational Intelligence Classification considers the hybridisation of AI models as a way to obtain AI systems that are capable of simulating aspects of Biological Intelligence. The IRIS classification shows how AI systems should be integrated with other computational systems and with the environment and finally the classification proposed by Medsker and Bailey defines five different ways of combining connectionist and symbolic AI systems from a practical point of view.

1.3 Combining CBR systems and Neural Networks

CBR systems have been successfully used in several domains: diagnosis, prediction, control and planning (López de Mántaras *et al.*, 1997). Although there are many successful applications based on just CBR technology, from an analysis of this type of system it appears that CBR systems can be successfully improved, combined or *augmented* (Hunt *et al.*, 1994) by other technologies. A hybrid CBR system may have a clearly identifiable reasoning process. This added reasoning process could be embedded in any of the stages that compose the CBR Cycle. For example the most common approaches to construct hybrid based CBR systems are:

- the CBR may work in parallel with a co-reasoner and a control module activates one or the other, i.e.: ROUTER (Goel, 1991),
- a co-reasoner may be used as a pre-processor for the CBR system as happens in the PANDA system (Roderman, 1993); and finally,
- a CBR may use the co-reasoner to augment one of its own reasoning processes (Corchado *et al.*, 1998).

The last approach is used in the majority of CBR hybrid systems. Hunt and Miles (1994) have investigated areas where Artificial Intelligence (AI) approaches (used as co-reasoners by this type of hybrid CBR based systems) are applied. Most early work in this area combined CBR systems with rule-based reasoning systems, but the number of applications in which other AI techniques are combined with case-based reasoning systems is increasing continually and quickly as has been reported by Medsker (1995), Sun and Alexandre (1997), and Lees (1999).

CBR systems are flexible systems capable of using the beneficial properties of other technologies to their advantage; in particular, the interest here is in the advantages of combining CBR and Artificial Neural Networks (ANN). During the last decade an increasing number of scientists have been researching into the hybridisation of CBR systems and ANNs. Before reviewing this area it is necessary to clearly define when and where ANN can be used in this context.

ANNs are not especially appropriate for stepwise expert reasoning and their explanation abilities are extremely weak. Nevertheless their learning and generalisation capabilities can be useful in many problems. Therefore they can only be used as part of CBR systems in those areas that do not involve knowledge explanation and reasoning. In particular, they can be used in areas involving knowledge generalisation. Learning is a powerful feature of most ANNs, and learning forms an intrinsic part of many stages of the CBR cycle, so ANNs can be used to learn to retrieve the closest case to a particular situation, or in other words to learn to identify the closest matching case. For an ANN it is reasonably easy in most situations to learn new cases and to learn how to generalise (adapt) a case from a pool of cases.

CBR systems and ANNs are complementary techniques, ANNs deal easily (and normally) with numeric data sets whereas CBR systems deal normally with symbolic knowledge. Even when symbolic knowledge can be transformed into numeric knowledge and numeric into symbolic, by doing this there is always the risk of losing accuracy and resolution in the data and hence obtaining misleading results. Therefore a combination of CBR systems and ANNs may avoid transforming data and therefore gain precision. As mentioned before, generalisation is a useful ability of most ANNs, but in many cases it is necessary to hold information about special cases, and this is a natural ability of CBR systems.

When CBR systems and ANN are used together, the most common approach (Reategui, 1996) is to hold the cases as an integral part of the ANN because CBR systems can successfully use them in the indexing and retrieval stages. For example, in the hybrid system created by Myllymaki and Tirri (1993), cases are identified as neurons of an ANN. The CBR system uses Bayesian probabilistic reasoning and is implemented as a

connectionist network (also called a belief network), which uses probability propagation to provide the theoretical explanation for the case matching process. Cases are represented as neurons in the middle layer of the ANN in this particular model.

Becker and Jazayeri (1989) have developed a hybrid system focused on design problems, in which cases are represented as neurons in the middle layer of an ANN and case-retrieval is done with a hybrid structure. Thrift (1989) uses an ANN with a back propagation learning algorithm for case filtering; the ANN selects the most relevant cases from the case base depending on some constrains (input to the ANN). GAL (Alpaydin, 1991) is based on a similar architecture, the difference being that GAL uses the prototype-based incremental principle, in which every class of objects is represented by the accumulation of relevant samples of the class and the modification of other class representations. Similar to a nearest-neighbour algorithm, this ANN grows when it learns and shrinks when it forgets because only representative cases are kept.

INSIDE (Lim *et al*, 1991) and ARN2 (Azcarraga *et al*, 1991) are very similar to GAL. In these systems, the neurons of the input layer of the ANN represents attributes, the nodes or neurons of the second layer correspond to prototypes (which are represented by n-dimensional vectors) and the neurons or nodes of the output layer represent classes. Each n-dimensional vector has an area of influence of a determined dimension. During learning the dimension of the areas of influence of the activated vector (prototype) is reduced if the ANN answer is wrong. Although INSIDE and ARN2 are very similar they differ in the methods that they use for learning the prototypes and adjusting their areas of influence.

The Prototype-Based Indexing System (PBIS) (Malek, 1995) was developed with the aim of improving the performance of the ARN2 model by keeping both prototypical and non-prototypical cases. PIBS has the memory divided to two levels. The first level is the middle layer of the ARN2 ANN and contains prototypical cases. The second level is a flat memory in which similar cases are grouped together in regions. Each region with similar cases is connected to the closest prototype of the same class.

PBIS also contains a region to store boundary cases that fall into uncertain areas. When a new case is presented to the ANN the prototype with the highest output is selected, if only one class is activated. When several classes are activated, the memory zones associated with the activated prototypes are selected and the most similar case is retrieved from these memory zones. If none of the prototypes are activated the system searches for similar cases in the atypical memory area.

Quan *et al*. (1994) have developed an algorithm for neural network based analogical case retrieval. This algorithm has been applied to industrial steam turbine design. Main *et al*. (1996) have investigated the use of fuzzy feature vectors and neural networks as a means of improving the indexing and retrieval steps in case-based reasoning systems.

PATDEX/2 (Richter *et al*, 1991) is a CBR-ANN hybrid system in which the relationship between the CBR and the ANN is different from the previous models. PATDEX/2 is a fault diagnosis system based on case-based reasoning technology. Cases are symptom vectors together with their associated diagnoses. In PATDEX/2, an ANN using a competitive learning algorithm is the core of the retrieval algorithm. The

similarity measure is based on a matrix that associates the relevance of every symptom to every possible diagnosis. The weights of this matrix are learned and modified by the ANN: after each diagnosis, the weights of the matrix are updated depending on the success of the diagnosis.

Garcia Lorenzo and Bello Perez (1996) use an ANN as a basis for calculating a measure of similarity between a new problem case and each stored candidate case. The ANN provides a mechanism to retrieve cases using information that in other models would require a parallel architecture. The connection between both case-based and rule-based reasoning mechanisms, and high-level connectionist models has been investigated by Sun (1996) in the process of exploring the use of such models for approximate common-sense reasoning.

Agre and Koprinska (1996) propose a different type of relationship between the CBR and the ANN in their hybrid model, which combines a CBR system and a knowledge-based ANN. The CBR is applied only for the correction of the knowledge-based ANN solutions that seems to be wrong. Potential corrections are carried out by matching the current situation against the cases that constitute the knowledge-based ANN training data set. Agree and Koprinska have shown that the performance of knowledge-based ANN (which are concerned with the use of domain knowledge to determine the initial structure of an ANN) can be considerably improved with the use of CBR systems.

Reategui *et al.* (1995, 1996) have been working on several hybrid ANN-CBR models and on general classifications of this type of system. Basically their hybrids are composed of two separate modules: a CBR system and an ANN. Both modules work independently; the reasoning process is interleaved between them and both co-operate via a central control unit. In one of Reategui's experiments, while the ANN learns general patterns of use and misuse of credit cards, the CBR system keeps track of credit card transactions carried out for a particular card (thus different sets of cases are used by the neural network and the CBR system). The central control mediates answers given by the two separate mechanisms.

In the domain of medical diagnosis, Reategui *et al*. (1996) have used an integrated CBR-ANN approach. The task of the neural network is to generate hypotheses and to guide the CBR mechanism in the search for a similar previous case that supports one of the hypotheses. The model has been used in developing a system for the diagnosis of congenital heart diseases. The hybrid system is capable of solving problems that cannot be solved by the ANN alone with a sufficient level of accuracy.

Liu and Yan (1997) have explored the use of a fuzzy logic-based ANN in a case-based system for diagnosing symptoms in electronic systems. The aim of the hybrid system is to overcome the problems related to the descriptions of uncertain and ambiguous symptoms.

Corchado *et al.* (1998) have also investigated the combination of CBR systems and supervised ANN. They have proposed an agent architecture for oceanographic forecasting in which the CBR agents and the ANN agents complement each other at different stages of the forecast. They have also developed a CBR model in which an ANN

automates the adaptation of cases, to solve a forecasting problem with high syntactical (numerical) connotations.

Following the work of Medsker and Bailey (1992) and inspecting the type of hybridisation used by the previously introduced authors, it may be appreciated that the two dominant models are **full integration**, in the form of a symbolic artificial neural network, and a model in which both components of the hybrid system are **totally or partially coupled**. In the latter case, most of the hybrid systems use an artificial neural network in the retrieval stage of the CBR cycle. In some systems both coprocessors are controlled by a meta-system and in other cases both coprocessors simply work in parallel doing independent tasks. The developers of the previously mentioned systems critically analyse the advantages and disadvantages of their models; in all cases the beneficial properties of the hybrids overcome their disadvantages. Studying this classification it is clear that there is a huge scope for investigating the combination of artificial neural networks with case-based reasoning systems. For example, different types of ANN can be used at different stages of the CBR life cycle to solve different problems.

ANNs have been used in the retrieval stage of a CBR system in situations in which there was no prior knowledge from constructing a KNN (k-nearest neighbour) algorithm or a Rule Based System (RBS). Although the use of ANN for retrieving cases has been shown to be successful (Mao *et al.*, 1994; Main *et al.*, 1996), it is not considered good practice if there is knowledge sufficient to build a KNN or a RBS. Also the real time constraints imposed by the nature of some problems must be taken into consideration to define whether or not it is possible to afford the time overhead for the training of the ANN.

Although the creation of neuro-symbolic models requires the existence of prototypical cases and a certain amount of knowledge, almost any CBR system can be represented as a neuro-symbolic system in which the neurons are prototypical cases or rule based systems. The connections between neurons could be defined also by rules. The following sections show how with a neuro-symbolic approach has been used to solve a complex oceanographic forecasting problem .

1.4 The Forecasting Problem

Oceans are dynamic habitats in which circulation is mainly driven by three external influences: (i) wind stress, (ii) heating and cooling, and (iii) evaporation and precipitation - all of which are, in turn, driven by radiation from the sun (Palmen *et al.*, 1969). The ocean circulation is what determines the mixture of water masses with different properties (such as temperature) and the variation of these properties with time in a particular geographical location. A water mass (or province) can be defined as a body of water with a common formation history. Oceans are in a continual state of flux (Tomczak *et al.*, 1994). Taken together, this bio-physical partitioning provides the descriptors of regional ecosystems or biogeochemical provinces, each with discrete boundaries and each having distinct

flora and fauna. In each of these provinces, the water properties are moderately homogenous and its variability can be described relatively easily. Our present knowledge of the ocean structure is still too weak to create a full model of its behaviour. Oceans are dynamic systems, in which the water masses are influenced by so many factors that it is extremely difficult to create even a partial model of the ocean. Therefore to develop a universal system for forecasting the temperature of the water ahead of an ongoing vessel is complicated.

Forecasting the structure of the water in such conditions is a difficult task due to the nature and behaviour of the ocean waters, the movement of which causes the water temperature to change in a complex manner (Tomczak *et al.*, 1994).

The forecasting task in such a complex environment requires the use of both historical data and the most recent real-time data available, thus enabling the forecasting mechanism to learn from past experiences in order to be able to predict, with sufficient confidence and accuracy, the values of desired parameters at some future point or points in time or distance.

Over the last few years researchers at the Plymouth Marine Laboratory (PML) and the University of Paisley have applied artificial intelligence methods to the problem of oceanographic forecasting. Several approaches have been investigated, both, supervised ANN (Corchado *et al.*, 1997) and unsupervised ANN (Corchado *et al.*, 1998) techniques have been investigated, as well as CBR and statistical techniques (Corchado *et al.*, 1998) with the aim of determining the most effective forecasting method. The results of these investigations suggest that, to obtain accurate forecasts in an environment in which the parameters are continually changing both temporally and spatially, an approach is required which is able to incorporate the strengths and abilities of several AI methods.

The problem of forecasting, which is currently being addressed, may be simply stated as follows:

> ***Given:*** a sequence of data values (which may be obtained either in real-time, or from stored records) relating to some physical parameter
>
> ***Predict:*** the value of that parameter at some future point(s) or time(s).

The raw data (on sea temperature, salinity, density and other physical characteristics of the ocean) which are measured in real time by sensors located on the vessel, consist of a number of discrete sampled values of a parameter in a time series. These data values are supplemented by additional data derived from satellite images, which are received weekly. In the present work the parameter used is the temperature of the water mass at a fixed depth. Values are sampled along a single horizontal dimension, thus forming a set of data points.

1.5 Hybrid CBR - Neural Network System

This section presents the hybrid system developed in this investigation. The hybrid system is composed of a *case-based reasoning* system and a *radial basis function artificial neural network*. It is a universal forecasting model. *Universal* in this context means the ability to produce accurate results anywhere in any ocean at any time. The system is capable of adapting itself in real-time to different oceanographic water masses.

To facilitate the understanding of the model this section focuses on the forecasting of the temperature of the water up to 5 km ahead. The concepts here presented are valid for longer distances; the metrics, dimensions of the vectors and some algorithms have been adapted for such longer distances as will be shown in following sections.

Figure 1 shows the top-level relationships between the processes comprising the hybrid CBR system. The cycle of operation is a derivation from the CBR cycle of Aamodt and Plaza (1994), and of Watson and Marir (1994). In Figure 1, shadowed boxes (together with the dotted arrows) represent the four steps of a typical CBR cycle; the arrows represent data coming in or out of the Case Base (situated in the centre of the diagram) and the text boxes represent the result obtained after each of the four stages of the cycle. Solid lines indicate data flow and dotted lines show the order in which the processes that take part in the life cycle are executed.

In the operational environment, oceanographic data (e.g. sea-surface temperature) is recorded in real time by sensors in the vessels; also, satellite pictures are received on a weekly basis. The satellite pictures are stored in a centralised database. A problem case is generated every 2 km using the temperatures recorded by the vessel during the last 40 km and consists of a vector of 40 temperature values, recorded at 1 km intervals. The problem case is used to retrieve the k most closely matching cases from the Case Base. Experiments carried out with data sets recorded in the Atlantic Ocean (cruise AMT 4) have shown that 40 data values at 1 km intervals was appropriate for the problem case (Rees *et al.*, 1997).

Each of the cases stored in the Case Base is defined by an *Input Vector* $(I_1, I_2, \ldots, I_{40})$ of water temperature values, a Forecast Value F (representing the value of the temperature of the water 5 km ahead of the point at which I_{40} was recorded) and several parameters defining its importance (how many times it has been retrieved, etc.) (refer to section 6.3). Both F and I_k must be recorded by a vessel following a straight line.

The k retrieved cases are adapted by a neural network during the reuse phase to obtain an initial (proposed) forecast. Through the revision process, the proposed solution is adjusted to generate the final forecast using error limits, which are calculated taking into account the accuracy of previous predictions. Learning (retaining) is achieved by storing the proposed forecast, modifying some parameters of the cases (as will be shown in following sections) and storing knowledge (ANN weights and centres) acquired by the ANN after the training and case adaptation.

Whilst Figure 1 presents the basic concepts of CBR as a cycle of operations, Figure 2 shows the detailed information flow throughout the CBR cycle and in particular how the

ANN has been integrated with the CBR operation to form a hybrid forecasting system. Data acquisition (top of Figure 2) is through sensors on board the vessel (in real time) and from satellite pictures (which are received weekly). The data is indexed so it can be transformed into cases and stored in the Case Base as required.

To obtain an accurate forecast in the vast and complex ocean it is imperative to use up-to-date satellite data. Fortunately, current technology now enables detailed satellite images of the oceans to be obtained on a weekly basis. The relevant data from these images is appropriately indexed for fast retrieval in a centralised database. Data is also acquired in real time as a vessel moves across the ocean; average sea surface temperatures are recorded every kilometre. Satellite images are used in this particular context because from them can be obtained the temperature of the water of the ocean in the form of thermal vectors. These thermal data vectors can be transformed into cases and stored in the case base (refer to Sections 1.5.1 and 1.5.3).
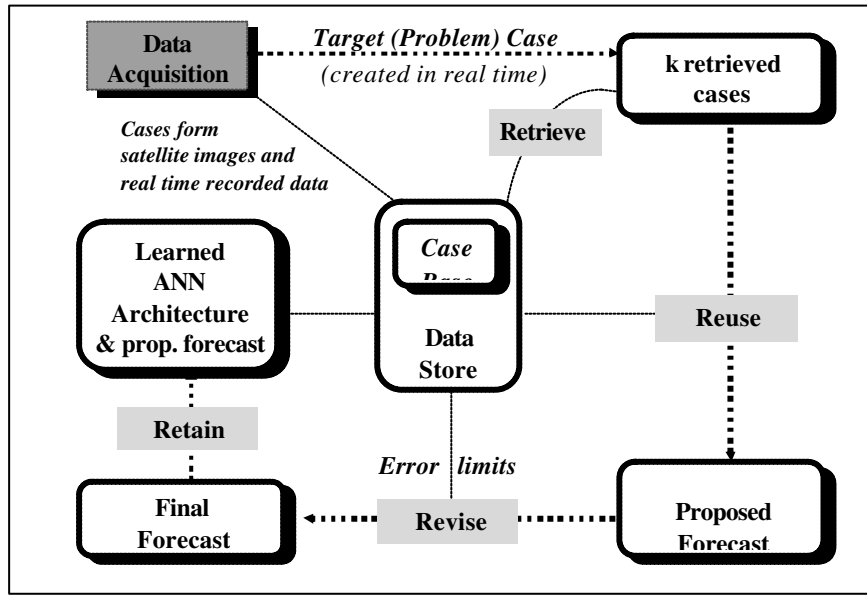

Figure 1: CBR skeleton.

During the retrieval phase, the k cases that most closely match the problem case are selected from the case base using k-Nearest Neighbour matching algorithms. These *k* cases are then used to compute the forecast value of the temperature of the ocean a constant distance ahead of the vessel. The set of k retrieved cases is used in the reuse phase of the CBR life cycle to train an ANN, the output of which is the Proposed Forecast (see Figure 2). The radial basis function ANN is retrained in real time to produce

the forecast; during this step the weights and the centres of the ANN, used in the previous prediction, are retrieved from the knowledge base and adapted, based on the new training set. The goal of the ANN is to extract a generalised solution from the k cases (refer to Section 1.5.5).

In the revise phase the Final Forecast is obtained by modifying the Proposed Forecast taking into account the accuracy of the previous predictions. Each case has an associated average error which is a measure of the average error in the previous predictions for which this case was used to train the ANN. The error limits are calculated by averaging the average error of each of the k cases used to train the ANN to produce the current forecast.
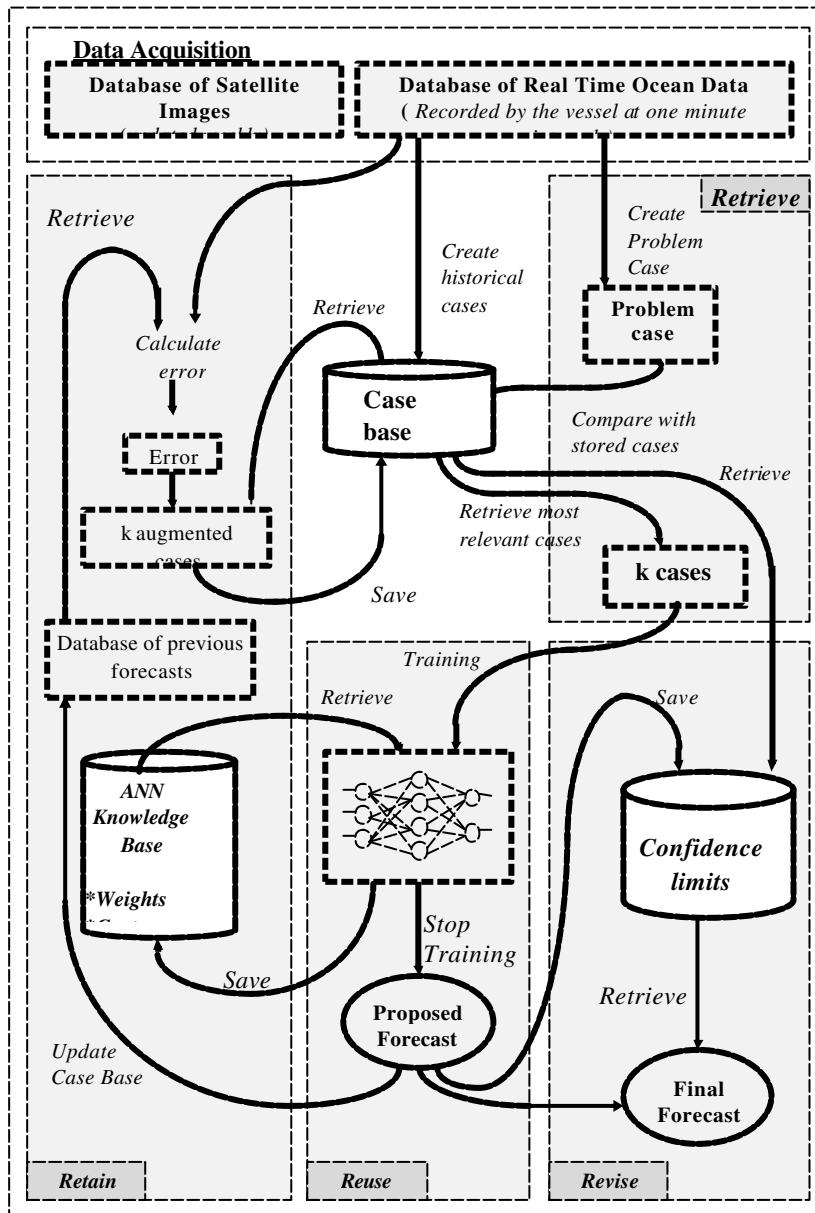
Figure 2: Hybrid system control flow.

Learning is achieved in two different ways:

1. after ***retraining the ANN***, by saving the internal structure of the ANN: i.e. the weights and centres. The ANN is continuously retrained and its internal structure is saved after each forecast is made,

2. by modifying some of the constituent parameters of the cases (as will be shown later).

A database records all the forecasts done during the last 5 km and all the cases used to train the ANN to obtain these forecasts. These forecasts are eventually compared with their corresponding real values of the temperature of the water (there is a 5 km lag between the point at which the forecast is made and the point for which the forecast is made). The forecasting errors are then used to modify relevant features of the cases, to prune the case base and to determine error limits, etc.

1.5.1 Case representation

A case represents specific knowledge about a particular situation. A case is created to represent the 'shape' of a set of temperature values (a vector of values) and the most representative characteristics of this vector. Each case is composed of the fields listed in Table 1.

A 40 km profile has been empirically found to give sufficient resolution (using representative data sets) to characterise the problem case (and the Input Vector I_k). The parametric features of the different water masses that comprise the various oceans vary substantially, not only geographically, but also seasonally. Because of these variations it is therefore inappropriate to attempt to maintain a case base with cases from all the water masses of the ocean. Furthermore:

- there is also no need to refer to cases representative of all the possible orientations that a vessel can take in a given water mass. Vessels normally proceed in a given predefined direction. So only cases corresponding to that particular orientation are normally required at any one time,

- recall also that the aim is to create a forecasting system which is able to adapt to the changes in the ocean environment, in time and space.

With the above considerations, the strategy adopted was to maintain a centralised data base in which all the thermal data tracks and satellite pictures, available for all the

water masses in the world, could be stored, in a condensed form; and then only retrieve from it (transformed into cases and stored in the case base) the data relevant to a particular geographical location.

Plymouth Marine Laboratory (PML) maintains a database composed of thousands of data profiles recorded during the last decade, together with satellite images. The database is updated weekly. For the purpose of the current research a new database has been constructed for a region of the Atlantic Ocean between the UK and the Falkland Islands (between latitudes: 50 to -52 and longitude: 0 to -60). This database is a subset of the main PML database.

Identification	Description
Identification	• Unique identification: positive integer number (between 0 and 64000).
Input Vector I_k	• A 40 km temperature input profile ($I_1, I_2, .., I_k$, where k = 40) representing the temperature of the water between the present position of the vessel and its position 40 km back.
Output Value: F	• a temperature value representing the water temperature 5km ahead of the present position of the vessel.
Source	• Data source from which the case was obtained (satellite image or data track). Its acquisition date, time and geographical co-ordinates identify each source. e.g.: SAT1501981030500-4910 i.e. a satellite picture taken on 15th January 1998, at 10:30 the top left corner having latitude 50.00º and longitude -49.10º.
Time	• Time when recorded. Although giving some redundancy of information, this field helps to ensure fast retrieval.
Date	• Date when the data was recorded (incorporated for the same reasons as for the previous field).
Geographical position	• The geographical co-ordinates of the location where the value I_{40} (of the Input vector) was recorded.
Retrieval	• The number of times the case has been retrieved to train the ANN (a non-negative integer).
Orientation	• An integer x ($1 \leq x \leq 12$) corresponding to the approximate direction of the data track, as indicated in Figure 6.3.
Retrieval time	• Time when the case was last retrieved.
Retrieval date	• Date when the case was last retrieved.
Retrieval location	• Geographical co-ordinates of the location in which the case was last retrieved.
Average error	• The average error in all forecasts for which the case has been used to train the ANN.

Table 1: Case attributes.

1.5.2 Case indexing mechanism

Time is a very important factor in real time forecasting. Therefore the indexing mechanism used to retrieve both the data stored in the database and the cases in the case base must be fast and reliable. Also the selection mechanism for the creation of cases from data stored in the database must be accurate. Research has also shown in the INRECA project (Wilke *et al.*, 1996) that very large case bases have poor performance results. Therefore only representative cases must be created and stored. This fact, together with the need for creating them within a small period of time, makes a good indexing algorithm essential.

<div align="center">(a)   (b)</div>
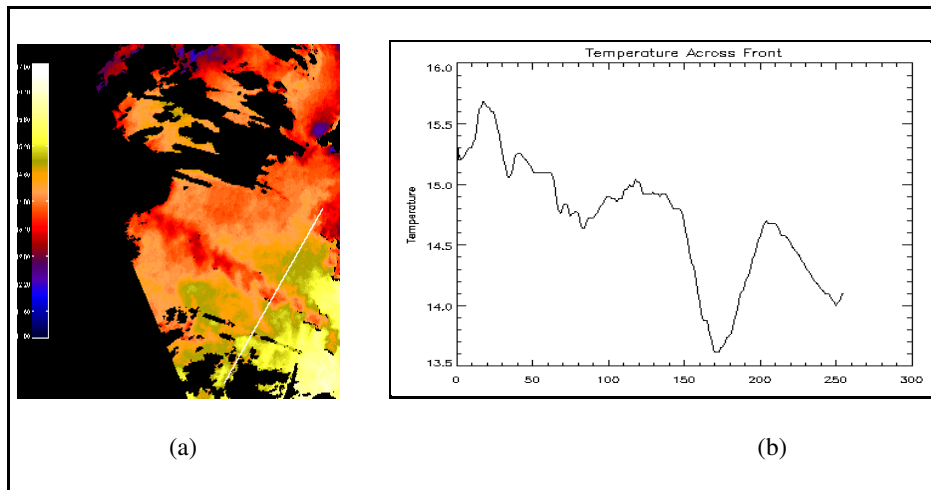
Figure 4: Satellite Image and a track obtained from a satellite image.

There are several approaches for organising the information held by a case base. Commonly used approaches are:
- **flat memory system**, that requires a large memory when the case base become large,
- **shared-features network**, that has a hierarchical case organisation (Kolodner, 1993), that requires a rigid network structure, hard to modify once the case base is in use and cases are added into the case base.

The complexity and the quantity of the data with which this system is dealing requires a simple but rigid indexing mechanism in order to minimise the retrieval time of the cases.

The relevant cases to be retrieved are those geographically close to the position of the vessel. The cases stored in the case base at any one time are always geographically close to the position of the vessel. This is enforced by the algorithms in charge of

retrieving data from the database and storing it into the case base. These algorithms also give priority to the retrieval of the most recent cases.

Because more refined spatial and temporal selection is required a simple indexing structure has been implemented which groups cases, taking into account their geographical proximity and secondly their temporal proximity. Cases within the same geographical region defined by a square of side 0.1 degrees (10 km) are stored together. This group is divided again into smaller units, each of which has been recorded in the same week.

1.5.3 Transformation of data into cases

Cases are constructed from the data held in the database and stored in the case base, according to the rules defined in Table 2.

Classification	Description
1	Cases within an area delimited by a circle of radius P km centred on the present position of the vessel. i.e. $P = X+(X*0.25)$. Where • X is the distance between the present position of the vessel and the *geographical position* of the case with a *retrieval* field equal to 4 and in which the *averaged error* is smaller than 0.05 and which has been retrieved within the last 20 km or 24 hours. If there is no case with a retrieval field equal to 4, the one having a value closest to 4 will be chosen. These threshold values have been empirically obtained in experiments carried out with data sets obtained in AMT (Atlantic Meridional Transect) cruises. • $25 \leq P \leq 200$
2	Cases with the same orientation as the present cruise track.
3	Cases from data recorded during the same month as the cases that are stored in the case base in which the forecasting error is less than 0.05 and which have been used during the last 24 hours or 50 km. Cases are also constructed from data recorded in the previous month under the same conditions.
4	Cases are constructed from data recorded in the last two weeks.

Table 2: Rules for case construction.

The classification of cases presented in Table 2 have been empirically obtained as a result of much experimentation (Corchado, 1999). The following section shows how cases are selected during the CBR operation to create the final output, how they are deleted from the Case Base and how they can be reused.

17

1.5.4 Case retrieval

From the data recorded in real time, the input profil*e, I,* of the problem case is created. A search is made in the case base to retrieve all cases having similar profiles. Five metrics are used to determine the similarity between the problem case and each of the retrieved cases.

At this stage the aim is to retrieve all the cases that are similar to the problem case. The cases stored in the case base have been extracted from satellite images or data tracks. Therefore it is required to recover as many cases similar to the problem case as possible so that in the following stage a forecasting model based on all the recovered cases may be created.

The metrics used in the retrieval process give priority to cases based on complementary criteria. They enable cases to be retrieved whose input profiles are similar to the problem case with respect to their temperature profiles (*Metric 1* and *Metric 2*), general trend in temperature (*Metric 3*), similarity in terms of the frequency of oscillation of the Sobel filter of the profile (*Metric 4*), and similarity with respect to the average sea temperature over the distance represented by the case (*Average Temperature Metric*).

  Metric 1

    This metric calculates the difference between I_{40} and eight other values of the *Input Profile (of the Problem Case)*, spread along the input profile at 5 km intervals (starting with I_1). This is repeated for all the cases stored in the Case Base. The value of the gradients are weighted, giving a higher value to the ones closer to I_{40}. The weighted gradient of each case (retrieved from the Case Base) is compared with the value of the weighted gradients of the *Input Profile of the Problem Case*, and the absolute value of the differences are added up. The value of Gradient 1 used to retrieve the cases is given by:

$$Metric\_1 = \sum_{i=0}^{7} \left(\left| \left(I_{40} - I_{i*5+1} \right) - \left(IA_{40} - IA_{i*5+1} \right) \right| * \left(84 + i*2 \right) / 100 \right)$$

    where the vector I_j, (j = 1, 2 … 40) represents the input profile of the problem case and IA_j, (j= 1, 2 …40) represents the input profile of the case being considered from the case base. The closer that the profiles being compared match, the smaller will be the value of *Metric 1*. This metric provides a general idea of the degree of similarity between the problem case and all the cases of the memory of the CBR.

  Metric 2

    This metric is similar to the previous one, the difference is that the input profile is first smoothed using a window of four values. This metric uses the difference between I_{40} and each of thirteen other values of the input profile of the problem case relating to points at 3 km intervals (starting at I_1). The values obtained are weighted and summed as in the calculation of *Metric 1*. This is repeated for all the cases stored in the case base. This metric gives a more general indication of the similarity between the present case and the retrieved cases than *Metric 1*. This metric provides a stronger degree of similarity than the previous one due to its higher level of

granularity and the fact that by smoothing the case vector, irrelevant features (such as noise) are eliminated. The smaller the value of the metric, the more similar is the retrieved case to the *Present Input Profile*.

Metric 3

The output of this metric is the absolute value of the difference between the gradient of the *Problem Input Profile* and each of the cases of the case base. The gradient is calculated using the average value from the first and last 20% of each *Input Profile*. A percentage of 20 has been empirically found to be an adequate value to calculate a gradient that defines, in general terms, whether the temperature is increasing or decreasing.

$$Metric\_3 = \left| \left(\left(\sum_{i=1}^{5} I_{i+35} - \sum_{i=1}^{5} I_i \right) / 5 \right) - \left(\left(\sum_{i=1}^{5} IA_{i+35} - \sum_{i=1}^{5} IA_i \right) / 5 \right) \right|$$

This metric compares the general trend of the problem case with the general trend of the retrieved cases, so for example, it can be identified cases which show a similar general increment or decrement in the temperature.

Metric 4

The *Sobel filter* (Gonzalez and Wintz, 1987) value is calculated for the present case and all the *input profiles* of the retrieved cases. The output of the metric 4 is the absolute value of the difference between the number of oscillations of the Sobel filter of the input profile of the retrieved cases and Problem case.

The value of the Sobel Filter for a case is calculated as follows:

$$(\forall x_i: 3 < i < 38): Sobel\_x_i = ((\sum_{j=i-2}^{i+2} x_j) - x_i) / 4$$

This metric gives priority to the cases, which Sobel filter is similar to the Sobel filter of the input vector of the problem case. This metric helps to identify cases from water masses of similar characteristics because case vectors extracted from similar water masses have similar Sobel filters (Corchado *et al.*, 1996).

Average Temperature Metric

The *Average Temperature Metric* compares the average temperature, over the distance represented by each retrieved case, with that of the problem case. This case is used to identify cases that have been extracted from the sea in similar seasons of the year and similar water masses because cases extracted from the same areas of the ocean and extracted during the same season have similar average temperatures.

After applying the above metrics to all the cases in the Case Base, the best matches to the problem case are used to obtain the final forecast. The best matches of each metric will be used to train a Radial Basis Function ANN in the adaptation stage of the Reuse phase. The number of best matches selected from the outcome of each metric is determined as follows:

1. For each metric, the value of the outcome is expressed on absolute scale between 0 and 1. Thus the cases, which are more similar to the problem case, will have a value closer to 0 and the more distant cases will have a value closer to 1.
2. For each metric, the two hundred best matches are used in the adaptation phase of the CBR cycle to train the ANN. If the value of the metric associated with any of these 200 cases is bigger than 3 times the value of the best match, this case is not used in the training of the ANN.

A reasonable number k of cases is required to train the ANN; empirically it has been observed that a value of between 500 and 1000 produces satisfactory results. If k is greater than 1000 it becomes impossible to train the ANN in the time available, whilst a value smaller than 500 may restrict the ANN's generalisation ability. The same cases may be selected using different metrics, and will have a higher influence during the adaptation step of the reuse phase.

The metrics presented above are only applied to the cases which have a *date* field equal to or within 2 weeks of the *date* field of any of the cases used to train the ANN in the most recent forecast, or for which the geographical position differs by less than 10 km to the geographical position of any of the cases used to train the ANN in the most recent forecast.

1.5.5 Case reuse (adaptation)

Several hybrid systems have been developed in which CBR components co-operate with one or more reasoning elements (Hunt *et al.,* 1994). In particular, there are a number of CBR systems that use Constraint Satisfaction, Numeric Constraint Satisfaction, Model Based Reasoning, etc., for case adaptation.

Case adaptation is one of the most problematic aspects of the CBR cycle. Most adaptation techniques are based on generalisation and refinement heuristics. This section proposes a novel approach based on ANNs and their ability to generalise. The ANN acts as a function that obtains the most representative solution from a number of cases. This ANN does not require any type of human intervention and has only a small number of rules that supervise the training of the ANN.

In the context of the present problem, instance-based reasoning (Aha *et al*., 1991) is required to compensate for the lack of guidance from specific and accurate background knowledge about the propagation of the temperature of the water of the oceans, with a relatively large number of instances. This is a highly syntactic CBR-approach, in the sense that a simple feature vector (refer to Section 1.5.1) is only needed to represent each instance and no user is required in the CBR life cycle.

Each of the cases or instances retrieved from the CBR represents a particular problem and its solution (feature vector). The aim of the CBR operation is to determine which of the cases stored in the CBR case base characterises better the present situation so that it may be reused. The determination of an algorithm to automate this process and retrieve the best case at any point is difficult because of the complexity of the environment, its dynamism and its heterogeneity.

The method adopted is to use a mechanism able to absorb the knowledge of all the cases that are representative of one particular problem and extrapolate from them a solution. To this end, experiments were carried out with nearest neighbour algorithms (which find the case among the retrieved cases that is most similar to the present problem), averaging techniques and artificial neural networks. A Radial Basis Function ANN has been found to be able to absorb the underlying model represented by the retrieved cases and generalise a solution from them, better than any other technique (Corchado *et al.*, 1997). This ANN is retrained before any forecast is made using the retrieved cases and the internal knowledge (weights and centres) of the Radial Basis Function ANN. Every time that the ANN is retrained, its internal architecture is adapted to the new problem and the cases are adapted to produce the solution, which is a generalisation of those cases.

Radial Basis Function ANNs are adequate in this problem because they can be trained fast, they have very good generalising abilities (though being better at interpolating than at extrapolating) (Corchado, 1999), and they learn without forgetting by adapting their internal structure (adding new centres) (Fritzke, 1994). This last property is particularly interesting in the present situation because since the ANN is continuously being retrained, it can learn new features within one particular water mass without forgetting a number of the others previously learned (for a variable number of training iterations).

Although this increases the training time, it improves the generalisation since at any time the forecast is not only based on the last k cases used to retrain the ANN, but also on those cases used in the more recent past which also influence the forecast; this contributes to the generation of a continuous, coherent and accurate forecast.

In the RBF network that has been built, cases are coded in order to create the input and output vectors used to train the ANN. The ANN uses 9 input neurons, between 20 and 35 neurons in the hidden layer and 1 neuron in the output layer. The input data is the difference between the last temperature (of the *Input Profile*) and the temperature values of the input profile taken every 4 km. Only one neuron is used in the output layer to forecast up to 5 km. The output is the difference between the temperature at the present point and the temperature 5 km ahead.

1.5.5.1 Centre and weight adaptation

Initially, 20 vectors are randomly chosen from the first training data set (composed of the retrieved cases), and are used as centres in the middle layer of the ANN. This number changes with training and the training data set determines it. The topology of the ANN (i.e.: number of neurones in each layer) have been empirically chosen after many tests with data sets extracted in the AMT cruises (Corchado *et al.*, 1996). The number of initial centres has been chosen taking into consideration the number of neurons in the input and the output layer.

All the centres are associated with a Gaussian function the width of which, for all the functions, is set to the mean value of the Euclidean distance between the two centres that are separated the most from each other.

The closest centre to each particular input vector is moved toward the input vector by a percentage α of the present distance between them. By using this technique the centres are positioned close to the highest densities of the input vector data set. The aim of this adaptation is to force the centres to be as close as possible to as many vectors from the input space as possible. An adaptation of this type is particularly important because of the high-dimensionallity of the input layer. α is initialised to 20 every time that the ANN is retrained, and its value is linearly decreased with the number of iterations until α becomes 0; then the ANN is trained for a number of iterations (between 10 and 30 iterations for the whole training data set, depending on the time left for the training) in order to obtain the best possible weights for the final value of the centres. The thresholds that determine the centres and weights adaptation have been empirically determined).

The delta rule (Bishop, 1995) is used to adapt the weighted connections from the centres to the output neurons. In particular, for each presented pair of input and desired output vectors, one adaptation step, according to the delta rule, is made.

1.5.5.2 Insertion of new units

A new centre is inserted into the network when the average error in the training data set does not fall more than 10% after 10 iterations (of the whole training set). In order to determine the most distant centre C, the Euclidean distance between each centre and each input vector is calculated and the centre whose distance from the input data vectors is largest is chosen. A new centre is inserted between C and the centre closest to it. Centres are also eliminated when they do not contribute much to the output of the ANN. Thus, a neuron is eliminated if the absolute value of the weight associated with a neuron is smaller than 20% of the average value of the absolute value, of the 5 smallest weights. The number of neurons in the middle layer is maintained above 20. This is a simple and efficient way of reducing the size of the ANN without dramatically decreasing its memory.

1.5.5.3 Termination of training

The ANN is trained for a maximum time of 2 minutes. In the real time operation the ANNs must produce a forecast every 2 km (6 minutes for a speed of 12 knots, which is the maximum speed that the vessel can attain). After that time the new set of training cases is retrieved and the ANNs are retrained. Therefore, even if the error is high the ANNs should produce a forecast. It has been found empirically, that these training times are sufficient to train the network and obtain a forecast with small errors (Corchado, 1999). At any point if the average error in the training data set is smaller or equal to 0.05 the training is stopped to prevent the ANN from memorising the training vectors. This threshold has

been chosen empirically. It has been shown that the ANN around this point stops the generalisation process and starts to learn and to memorise the training vectors.

1.5.6 Case revision

After case adaptation a crisp value is obtained for the forecasted temperature 5 km ahead of the vessel. This value is rarely 100% accurate, therefore revision is required to obtain a more realistic output.

Since this is a real-time problem it is not possible to evaluate the outcome of the system before it is used. The solution to this problem is to define error limits, which will substitute the crisp output with a band or error interval around the output of the ANN. If the error limits are too wide the forecast will be meaningless; therefore a trade off is made between a broad error limit (that will guarantee that the real solution is always within its bands) and the crisp solution.

The expected accuracy of a prediction depends on two elements: the water mass in which the forecast is required and the relevance of the cases stored in the case base for that particular prediction.

Each water mass has been assigned a default error limit CL_0, which has been empirically obtained. Every time a cruise crosses a water mass, a new error limit CL_z (where $0<z<6$) is calculated by averaging the error in all the predictions made. If, for a certain water mass, z is equal to 5, and a vessel crosses it again, the older CL is substituted by a new one. Therefore there are at most 5 error limits associated to a water mass. This number is not critical, a smaller value can also guarantee stability in such error limits, and a larger number does not provide a better result. The CL_z error limits are used in collaboration with the *average error* of the cases used to train the ANN for a given forecast. The error limit determines the interval centred in the crisp temperature value, obtained by the ANN, for which there is a high probability that the forecast is within this interval. The value of the probability varies deepening on the distance of the forecast, but must be higher than 90%. Then, if the output of the ANN is F, the average value of the accumulated errors of the cases taking part in a given forecast is AE and ACL is the average value of the CL_z error limits, the error interval is defined by

$$[F - ((AE*0.65)+(ACL*0.35)), F + ((AE*0.65)+(ACL*0.35))]$$

The values used in this formula have been empirically obtained using a sufficient amount of data from all the water masses of the Atlantic Ocean. However, these values may not be appropriate for water masses from different oceans.

1.5.7 Case retention and learning

Incorporating into the case base what has been learned from the new prediction is the last stage of the CBR life cycle. Learning is achieved in different ways in the system. When the ship has travelled a distance of 5 km (on a straight course) after making a

forecast, it is possible to determine the accuracy of that forecast, since the actual value of the parameter for which the forecast was made is then known. This forecasting error is used to update the average error attribute of the cases used in that particular forecast. The cumulative error field of the cases used to train the neural network is continuously being updated and contributes to the learning strategy of the system. Accurate error limits are obtained only if the average error attribute of the cases is modified in this way.

Pruning the case base also contributes to the learning; cases in which average error attribute is very high are removed. The maximum permissible average error needs to be defined. Empirically it has been found that for cases in which the average error attains a value 0.12, the average error never subsequently reduces to a value smaller than 0.05. Therefore a threshold of 0.1 in the *average error* was chosen to determine which cases must be deleted. If the average error of a case is equal to or higher than 0.1, the case is removed from the case base. Furthermore, cases which have not been used during the previous 48 hours are deleted; so also are cases which have not been used in the previous 100 km.

It is necessary to determine when the case base must be updated with additional cases from the database. This is done when the database receives new satellite images (once per week). If the forecasting error is higher than 0.2 for more than 20 predictions, additional cases are created from data stored in the database. This is a measure used to include fresh cases in the case base; this helps to reduce the forecasting error.

If, over a period of operation in a particular water mass, it is found that most of the cases selected from the case base are clustered around some point a distance x, say, either ahead or behind the vessel, this suggests that the whole water mass has moved this distance x since the data, from which the cases were created, were obtained. In such a situation, the operational strategy is then to utilise cases relating to this indicated area, which is centred on a position a distance x from the current position.

The modification and storage of the internal structure of the ANN contribute substantially to the learning of the system. The weights and centres of the network, and also the width of the Gaussian functions associated with the centres, are modified during the adaptation process and stored in the network knowledge base.

Learning is a continuous process in which the neural network acts as a mechanism that generalises from the input data profiles and learns to identify new patterns without forgetting previously learned ones. The case base may be considered as a *long term memory* since it is able to maintain a huge number of cases that characterise previously encountered situations.

In contrast, the network knowledge base may be considered to behave as a *short term memory* that has the ability to recognise recently learned patterns (i.e. the knowledge stored in the internal architecture of the neural network) that enable it to adapt the system to cope with localised situations.

1.6 Results and Discussion

This chapter has described the hybrid system developed to forecast in real-time the temperature of the water ahead of an ongoing vessel. The hybrid system is composed of a case-based reasoning system and an artificial neural network. The ANN assists the CBR system during the adaptation of cases and also contributes to the learning of the system.

The hybrid system holds in its memory a huge amount data relating to forecasting events and selects from it those that are the most similar to the new forecasting situation. The ANN uses the retrieved cases to create a model, in real time, for a particular water mass. This reasoning model makes use of the most up-to-date data to generate a solution in real-time in order to overcome the difficulties of predicting the evolution of a dynamic system in real time. Although the present chapter has focused on the prediction of the temperature 5 km ahead, the same strategy has been used to forecast up to 20 km. A complete analysis of the results obtained with this system may be found in Corchado (1999).

The approach presented here combines the advantages of both connectionist and symbolic AI. The hybrid system has been tested in the Atlantic Ocean in September 1997 on a research cruise from the UK to the Falkland Islands. The cruise crossed several water masses and oceanographic fronts. The obtained results were very encouraging and indicate that the hybrid system is able to produce a more accurate forecast than any of the other technique used in this experiment.

Although the experiment has been carried out with a limited data set (over a distance of 11000 km between the latitudes 50º North and 50º South), eleven water masses with different characteristics were traversed, six of them containing fronts; the Falkland Front, in particular, is one of the most chaotic oceanographic areas in the world. It is believed that these results are sufficiently significant to be extrapolated to the whole Atlantic Ocean.

Prediction Distance (km)	Hybrid system Average error (ºC)	Hybrid system: % of inadmissible predictions
5	0.020	2.6
10	0.048	6.2
15	0.132	8.1
20	0.260	7.5

Table 3: Average error and percentage of inadmissible predictions with the hybrid system.

Table 3 shows the results obtained with the hybrid system when forecasting the water temperature at different distances. The table also show that the percentage of inadmissible predictions (those that show that the temperature of the water is raising when it is decreasing and vice versa) is always smaller than 10%. This percentage is above this limit when forecasting further ahead than 20 km.

The forecasting ability of the system is highest in areas with small instabilities and where there are many data profiles from which to choose in the retrieval stage of the CBR cycle. The forecast is less accurate in areas with large changes and many instabilities. The system is not able to forecast if there are no data profiles in the region where the forecast is made.

Experiments have also been carried out to evaluate the performance of the hybrid forecasting approach in comparison with several separate neural networks and statistical forecasting methods (Corchado *et al.*, 1998): a Finite Impulse Response (FIR) model, an RBF network alone (trained with the data recorded during the 160 km previous to the forecast point), a linear regression model, an Auto-Regressive Integrated Moving Average (ARIMA) model and a CBR system alone (using the cases generated during the 160 km preceding the forecast point). Table 4 shows the average error in the forecast using these methods.

Prediction Distance (km)	Forecasting Method						
	FIR4*4 (ºC)	FIR8*5 (ºC)	RBF (ºC)	Linear regression (ºC)	ARIMA (ºC)	Specialised RBF (ºC)	CBR (ºC)
5	0.099	0.096	0.114	0.174	0.129	0.034	0.12
10	0.206	0.192	0.226	0.275	0.231	0.076	-
15	0.343	0.324	0.351	0.429	0.372	0.144	-
20	0.468	0.435	0.469	0.529	0.471	0.223	-

Table 4: Average forecasting error with different methods.

The success of the system in generating effective forecasts may be attributed to the combination of an extensive database of past cases, supported by the neural adaptation mechanism which, each time around the forecasting cycle, enables the forecasting process to learn from all the selected closely matching cases.

The experimental results obtained to date indicate that the neural network supported case-based reasoning approach is effective in the task of predicting the future oceanographic parameter values. Extrapolating beyond these results, it is believed that the approach may be applicable to the problem of parametric forecasting in other complex domains using sampled time series data.

Acknowledgement

References

Aamodt A. and Plaza E. (1994). Case-Based Reasoning: foundational Issues, Methodological Variations, and System Approaches. AICOM. Vol. 7. No 1. March 1994. pp. 39-59.

Agre, G. and Koprinska, I. (1996). Case-based refinement of knowledge-based neural networks. In J. Albus, A. Meystel and R. Quintero (Eds.) Proceedings of the International Conference on Intelligent Systems: A Semiotic Perspective (Volume II).

Aha D., Kibler D. and Albert M. K. (1991). Instance-Based Learning Algorithms. Machine Learning. Vol. 6 (1). pp 37-66.

Alpaydin G. (1991). Networks that grow when they learn and shrink when they forget. Technical Report TR 91-032. International Computer Science Institute. May, 1991.

Azcarraga A. and Giacometti A. (1991). A prototype-based incremental network model for classification tasks. Fourth International Conference on Neural Networks and their Applications. Nimes, France. 1991.

Bezdek J. C. and Jazayeri K. (1989). A connectionist approach to case-based reasoning. K. J. Hammond, Ed. Proceedings of the Case-Based Reasoning Workshop. Pensacola Beach, Florida. Morgan Kaufmann. pp. 213-217

Bezdek J. C. (1994). "What is Computational Intelligence?," in Computational Intelligence: Imitating Life. IEEE Press. Nueva York. Ed. Zurada J. M., Marks II R. J. and Robinson C. J. pp. 1-12.

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

Corchado J. M. (1995). Application of Filters, Masks and Artificial Neural Networks to Oceanographic Frontal Warning, BSc Honours Dissertation, University of Paisley.

Corchado J. M., Rees N. and Aiken J. (1996). Internal Report on Supervised ANNs and Oceanographic Forecasting. PML, 1996. Plymouth, UK. 30, December.

Corchado, J., Lees, B., Fyfe, C. and N Rees, N. (1997). Adaptive agents: learning from the past and forecasting the future, Procs PADD97 - First International Conference on the Practical Application of Knowledge Discovery and Data Mining. London, 23rd to 25th April, 1997, Practical Application Co. Ltd., pp 109-123.

Corchado J. M., Lees B., Fyfe C., Rees N. and Aiken J. (1998). Neuro-Adaptation Method for a Case Based Reasoning System. International Joint Conference on Neural Networks. Anchorage, Alaska, USA. May 4-9.

Corchado, J. (1999) Neuro-Symbolic Model for real-time forecasting problems. Ph.D dissertation, University of Paisley, UK.

Fritzke B. (1994). Fast Learning with Incremental RBF Networks. Neural Processing Letters. Vol. 1. No 1. pp. 2-5.

García Lorenzo M. M. and Bello Pérez R. E. (1996). Model and its different applications to case-based reasoning. Knowledge-Based Systems. November. Vol. 9. No 7. pp. 465-473. Elsevier Science B.V. Amsterdam, Holand.

Garcia Lorenzo M.M. and Bello Perez R.E. (1996) Model and its different applications to case-based reasoning, Knowledge-Based Systems, Nov, Vol.9, No.7, pp.465-473 Elsevier Science B.V., Amsterdam, Netherlands.

Goodman, M. (1994). Results on controlling action with projective visualization. In Proceedings of the Twelfth National Conferenc on Artificial Intelligence, pp. 1245-1250, Seattle, WA: AAAI Press.

Hunt J. and Miles R. (1994). Hybrid case-based reasoning. The Knowledge Engineering Review. Vol. 9:4. pp. 383-397.

Krovvidy, S. and Wee, W.G. (1993). Wastewater treatment systems from case-based reasoning, Machine Learning, Vol. 10, pp. 341-363.

Lees, B., Rees, N. and Aiken, J. (1992). Knowledge-based oceanographic data analysis, Procs. Expersys-92, editors: F Attia, A Flory, S Hashemi, G Gouarderes, J Marciano, IITT International Paris, October 1992, pp. 561-65.

Lees B., Kumar, B., Mathew, A., Corchado, J., Sinha, B. and Pedreschi, R. (1998). A Hybrid Case-Based Neural Network Approach to Scientific and Engineering Data Analysis. In Applications and Innovations in Expert Systems VI, editors: R. Milne, A, Macintosh and M. Bramer, Springer.

Lees, B. (1999) Hybrid case-based reasoning: an overview. Int. Workshop on Hybrid CBR systems, ICCBR 99, Munich, July, 1999.

Lim H. C., Lui A., Tan H. and Teh H. H.(1991). A connectionist case-based diagnostic expert system that learns incrementally. Proceedings of the International Joint Conference on Neural Networks. pp. 1693-1698. 1991.

Liu Z.Q. and Yan F. (1997). Fuzzy neural network in case-based diagnostic system IEEE Transactions on Fuzzy Systems, May 1997, Vol.5, No.2, pp.209-222.

López de Mántaras R. and Plaza E. (1997). Case-Based Reasoning: an overview. AI Communications 10. pp. 21-29. IOS Press.

Main J, Dillon T.S. and Khosla R. (1996). Use of fuzzy feature vectors and neural networks for case retrieval in case based systems, Procs. Biennial Conference of the North American Fuzzy Information Processing Society - NAFIPS, pp.438-443 IEEE, Piscataway, NJ, USA.

Malek M. (1995). A connectionist indexing approach for CBR systems.. Veloso M and Aamodt A., Ed.. Case-Based Reasoning: Research and Development. First International Conference, ICCBR-95. pp. 520-527. Sesimbra, Portugal. Springer Verlag. 1995.

Mao Q., Qin J., Zhang X. and Zhou J. (1994). Case prototype based design: philosophy and implementation. Proceedings. Computers in Engineering 1994. Vol. 1. pp. 369-74. Ed. Ishii K. September 11-14. Minneapolis, MN, USA.

Medsker L. R. and Bailey D. L.(1992). Models and guidelines for integrating expert systems and neural networks. Hybrid Architectures for Intelligent Systems. Ed. Kandel A. and Langholz G. CRC Press, Boca Raton.

Medsker L. R. (1995). Hybrid Intelligent Systems. Kluwer Academic Publishers.

Palmen E. and Newton C.W. (1969). Atmospheric Circulations Systems. Academic Press. Pp. 602.

Reategui E.B., Campbell J.A. and Leao B.F. (1997) Combining a neural network with case-based reasoning in a diagnostic system Artificial Intelligence in Medicine, Jan, Vol.9, No.1, pp.5-27, Elsevier Science B.V., Amsterdam, Netherlands.

28

Rumelhart, D. E and McClelland, J. L. (1996). Parallel Distributed Processing, Volume 1: Foundations, MIT Press.

Quan Mao; Jing Qin; Xinfang Zhang and Ji Zhou. (1994). Case prototype based design: philosophy and implementation Procs Computers in Engineering 1994. p. 369-74 vol.1, edited by Ishii, K. 11-14 Sept., Minneapolis, MN, USA, ASME New York, NY, USA.

Reategui E. B. and Campbell J. A. (1995). A classification system for credit card transactions. Haton J.P. , Keane M. and Manago M., Ed. Advances in Case-Based Reasoning: Second European Workshop. EWCBR-94. pp. 280-291. Chantilly, France. Springer Verlag.

Reategui E. B., Campbell J. A. and Leao B. F. (1996). Combining a neural network with case-based reasoning in a diagnostic system. Artificial Intelligence in Medicine. January. Vol. 9. No 1. pp.5-27. Elsevier Science B.V. Amsterdam, Holand.

Rees N., Aiken J. and Corchado J. M. (1997). Internal Report: STEB Implementation. PML, Plymouth, UK. 30, September.

Richter A. M. and Weiss S. (1991). Similarity, uncertainty and case-based reasoning in PATDEX. Boyer R.S., Ed. Automated reasoning. Kluwer. pp. 249-265.

Soucek B. and IRIS group (1991). Neural and Intelligent Systems Integration. John Wiley & Sons, New York.

Sun, R. (1996). Commonsense reasoning with rules, cases, and connectionist models: a paradigmatic comparison, Fuzzy Sets and Systems, Vol: 82 No: 2 pp. 187-200 Elsevier, Sept. 1996.

Sun R. (1996). Commonsense reasoning with rules, cases, and connectionist models: a paradigmatic comp arison. Fuzzy Sets and Systems. Vol. 82. No 2. pp. 187-200. Elsevier, September. 1996.

Sun R. and Alexandre F. (1997). Connnectionist-symbolic integration: from unified to hybrid approaches. Lawrence Erlbaum Associates.

Thrift P. (1989). A neural network model for case-based reasoning. Hammond K. J., Ed. Proceedings of the Case-Based Reasoning Workshop. pp. 334-337. Pensacola Beach, Florida. 1989. Morgan Kaufmann.

Tomczak M. and Godfrey J. S. (1994). Regional Oceanography: An Introduction. Pergamon. New York. 1994. ISBN: 0080410219.

Watson I. and Marir F. (1994). Case-Based Reasoning: A Review. Cambridge University Press, 1994. The knowledge Engineering Review. Vol. 9. No 3.

Wilke W. and Bergmann R. (1996). Incremental Adaptation with the INRECA - System. ECAI 1996 Workshop on Adaptation in Case-Based Reasoning.