WIND AMBIGUITY REMOVAL BY THE USE OF NEURAL NETWORK TECHNIQUES

by

F. BADRAN*,**, S. THIRIA*,**, and M. CREPON***

* Laboratoire de Recherche en Informatique

Université de Paris Sud, B 490 - 91405 ORSAY Cedex (FRANCE)

** CEDRIC, Conservatoire National des Arts et Métiers

292 rue Saint Martin - 75003 PARIS (FRANCE)

*** Laboratoire d'Océanographie et de Climatologie (LODYC)

T14 Université de PARIS 6 - 75005 PARIS (FRANCE)

Abstract

This paper deals with the removal of ambiguities existing on the direction of the wind measured by satellite scatterometers. It is shown that neural networks are well adapted to solve this problem. Results using ERS.1 simulated scatterrometric wind fields are presented. It is found that 99% of the ambiguities can be removed when imposing a rate of 25% ambiguity at 180° and 90°.

WIND AMBIGUITY REMOVAL BY THE USE OF NEURAL NETWORK TECHNIQUES

by

F. BADRAN*,**, S. THIRIA*,**, and M. CREPON***

* Laboratoire de Recherche en Informatique

Université de Paris Sud, B 490 - 91405 ORSAY Cedex

** CEDRIC, Conservatoire National des Arts et Métiers

292 rue Saint Martin - 75003 PARIS

*** Laboratoire d'Océanographie et de Climatologie (LODYC)

T14 Université de PARIS 6 - 75005 PARIS (FRANCE)

INTRODUCTION

By the 1990's several scatterometers are going to fly on board satellites dedicated to earth and ocean observation. The European ERS.1 AMI (Advanced Microwave Imager) is planned to launch in 1991 and the US NSCAT (NASA SCATterometer) is scheduled to launch in 1995 on the Japanese satellite ADEOS. These scatterometers will provide wind vectors on a grid mesh of 50*50 km with a time and space coverage which will be dramatically improved with respect to conventional means of observation. For the first time oceanographers can expect to obtain an adequate description of the forcing of ocean circulation which depends on the wind-stress vector for the surface layers and on the wind-stress curl for the large scale and deep motions.

Scatterometers are active microwave radar which accurately measure the power of transmitted and backscatter signal radiation in order to calculate the normalized radar cross section (σ_0) of the ocean surface. The σ_0 depends on the wind speed, polarization, and incidence angle which is the angle between the radar beam and the vertical at the illuminated cell and the azimuth angle which is

the horizontal angle χ between the wind and the antenna of the radar (Freilich and Chelton, 1986). The physics of the interaction of the radar beam with a rough sea-surface is poorly understood. Many effects as wave breaking, modulation by long waves, and the effects of rain make the problem complicated. An empirically based relationship between σ_0 and the local wind vector can be established which leads to the determination of a geophysical model function

Since it is found that the σ_0 varies harmonically with χ, (the backscatter maxima is at upwind and downwind and its minima at crosswind angle), it becomes possible to compute the wind direction by using several antennas with different orientations with respect to the satellite track. The geophysical model function can then be inverted and the wind vector computed. The ERS.1 and N.SCAT scatterometers have three antennas oriented in different directions. In the absence of noise, the determination of the wind direction would be unique in most cases. Noise-free ambiguities arise due to the bi-harmonic nature of the model function with respect to χ which is in the form $\sigma_0 = \cos(2\chi)$. It results in two major ambiguities which are approximately 180° apart. In practice, since the backscatter signal is noisy the number and the frequency of ambiguities is increased.

Several techniques have been suggested to remove these ambiguities. Some of them deal with large sets of data considered as a whole and use global techniques as objective analysis (Levy and Brown, 1986), assimilation into meteorological models (Atlas, 1984), or variational methods (Hoffman, 1982, 1984; Roquet and Ratier, 1988). Others deal with local neighborhood properties as the Bayesian statistical approach of K.J. Schwenzfeger (1985), the heuristic dealiasing method of CREO (A. Cavanié and D. Offiler, 1986) and the Median filter (Schultz, 1990).

In the present paper we propose a new approach based on concepts developed in the context of Artificial Intelligence. The method used to remove ambiguities from scatterometer wind fields is related to connectionist models which are supervised learning techniques and to image processing (Fogelman-Soulié et al, 1986, 1987; Kohonen, 1987; Lippmann, 1987). Recently, connectionist models have been recognized as an alternative tool to solve difficult problems of signal , speech or image analysis [Bourlard and Wellekens, 1989; Aarts and Korst, 1989; R. Azencott, 1990]. Such techniques have never been used in remote sensing, so one of the goal of this work is to show their usefulness in that field by solving a specific problem.

In the present problem, the method assumes the existence of a consistent topological relationship among wind vectors located in some neighborhood. Its conceptual framework is to determine the optimum topological filter to dealiase the wind direction. First, during a learning phase, the weights of this generalized filter which represent the connections of the neural network used are computed automatically from specific situations where the actual wind vectors are known. Then the network is applied to the observed wind field in order to remove ambiguities, using an image processing technique.

A brief review on neural networks and the specific model used is presented in Section 1. Details about the application of the method to remove scatterometer wind ambiguities are given in Section 2. Then in Section 3 the method is applied to a simulated wind field obtained from the fine grid mesh meteorological model of the ECMWF (European Center for Medium-range Weather Forecast). The validity of the method and its comparison to other methods are developed in Section 4

1 - AUTOMATA NETWORK

Connectionist models are based on the use of **automata networks**, which are large sets of many interconnected simple elements or automata. The complexity of their internal structure allows one to memorize and retrieve information. In the present section, a brief description of automata networks is presented

1.1 Preliminaries.

An **automaton** (or neuron) is an element defined by an **internal state** $s \in S$ and a **transition function** f which allows it to change its state. As an example, the state space S might be

{0,1} where 0 would be interpreted as an inactive state and 1 as an active state. The space S might also be the two grey levels in an image, a set of discrete values {0,1,…,k} for an image having k+1 grey levels or even a continuum [-1,1].

The transition function f is a function of the states of the different automata which are connected to the automaton whose state is computed. The automaton i which is in state $s_i \in S$, receives **n** signals $s_1, \ldots, s_n$ from the **n** connected automata, and will get a new state s'_i computed as:

$$s'_i = f(s_1, \ldots, s_i, \ldots. s_n)$$

Example: An automaton realizes the AND function, when receiving two signals s_1 and s_2, its internal state s gets

$$s' = 1 \quad \text{if } s_1 = s_2 = 1$$

$$s' = 0 \quad \text{otherwise}$$

An **automata network** is a set of interconnected automata. Each automaton receives signals only from the automata to which it is connected (or from the environment) (Fig. 1). In the following, it is assumed that all automata :

- have the same state space S and that the signals sent from an automata to other automata are the states of the automata only.
- are quasi-linear**.**

In such a case $S = \mathbb{R}$ ($\mathbb{R}$ is the set of real numbers) and the transition function f of the automaton i is a non-linear function of the **total input** A_i received by automaton i from the connected automata (Fig 1a). The transition function is of the form:

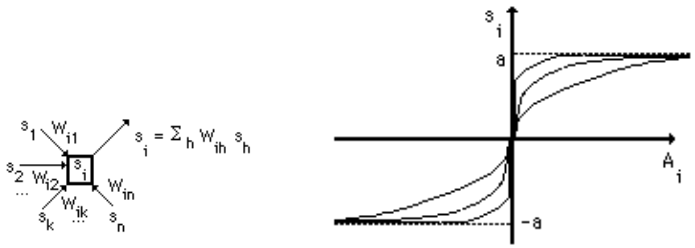
$$s_i = f(A_i) \qquad \text{with} \qquad A_i = \sum_h w_{ih}\, s_h \tag{1}$$

where w_{ij} is called the **connection weight** of the connection from j to i. They are real numbers allowed to weigh the various influences of the connected automata. They can be adapted by a learning process so that the network realizes a definite task.

Most often the non-linear function f is the sigmoid function :

$$f(u) = a\,[\, e^{\alpha u} - 1\,] \,/\, [\, e^{\alpha u} + 1\,]$$

The parameter $T = 1/\alpha$ is called temperature by analogy with the energy function used in many problems occurring in theoretical physics. Varying T from a high value to zero enables the refinement of the network's accuracy (see Fig 1b). With such a function the computed state of the neuron is a continuously valued real number between [-a, +a].



(a) (b)

Figure 1: (a) Diagram of an automata (neuron).

(b) Sigmoid function f for different values of α . The figure shows different functions f for T varying from 4 to 1 (the smallest T , the steepest the curve)

The pattern of the connections of the automata induces specific properties. In the present time, most automata networks used for applications are Multi-layered Networks formed by several layers of automata. One of the main interests of these automata networks is that they can simulate a large variety of functions.

1.2 The Quasi-linear Multi-layered Networks (Q.M.N)

Multi-layered architecture has one layer receiving the inputs, one layer which broadcasts output and one or more intermediate layers (the hidden layers) which are situated in between. In the network utilized, connections can only go from lower layers to higher layers, and no connection links cells within the same layer (Fig. 2).

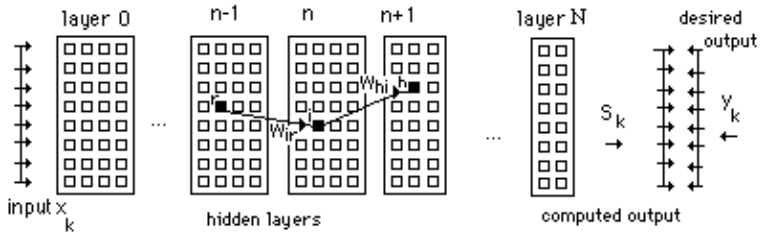


Figure 2 : Multi-layered network

x_k represents the input vector, s_k the output computed by the network and y_k the actual output to which s_k must converge

Such an architecture is very suitable to approximate multidimensional functions and to solve classification tasks. In fact, multi-layered networks provide adaptive systems allowing the design of transfer functions (or associative memories) [Funahashi, 1989]. The association could be between sets of patterns or real numbers or more generally between m vectors $(x_1,\ldots,x_m)$ in $\leftarrow^n$ and m corresponding vector $(y_1,\ldots,y_m)$ in $\leftarrow^p$ (Fig 3).

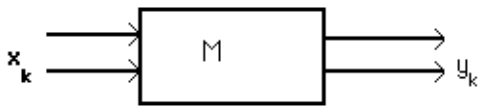


Figure3: Associative memory.

An input x_k is presented to a machine M (the Multi-layered-Network). The transfer function F associated with M allows the computation of an output y_k associated with the input x_k

Neural networks are made up of two classes of parameters :

- Some are dependent on the expert who built the network. These parameters are related to the network architecture (number of layers, number of automata in each layer, topological connections of automata, "temperature"). They are problem-dependent and must be adapted to the study faced (range of variations of the output for example), but cannot be learned. The expert plays an important role in fixing them by taking into account the physics of the problem and empirical considerations.
- Others as the weights w_{ij} of the connections must be learned from a learning set . These weights are determined in order to perform the optimal association. In the present work, the w_{ij} are computed by using the gradient-back-propagation algorithm [Hinton,1987] during a learning phase. This algorithm uses actual data in the form of associations (x_k, y_k) given by the "expert" in the field studied.

1.3 The Gradient-back-propagation algorithm (GBP algorithm)

As soon as the architecture is chosen (number of layers, number of automata by layers, the parameter T), the network is trained to realize the associations $(x_k \varnothing y_k, (k = 1\ldots m)$). The dimension n of the input layer (number of automata) is set to that of the x_k vectors and the dimension p of the output layer to that of the y_k.

First, the connection weights are initialized to random values. Then an input x_k is presented to the network. The states of the n automata on the input layer are forced at the x_k values; these states are propagated through the different layers of the network by using rule (1). The states s_k of the last layer is called the computed output associated with x_k. It is compared to the desired output y_k and an associated cost is defined as:

$$C_k = \| s_k - y_k \|^2 = \sum_j [s_{jk} - y_{jk}]^2$$

where j indices the p automata in the last layer.

The total cost is :

$$C(W) = \sum_k C_k \quad (k = 1\ldots m).$$

The total cost C is then minimized using a stochastic gradient method, the so-called Widrow Hoff rule [Widrow, 1960, 1985]. Each step minimizes the error on one association only and the weights are modified according to equation (2). Vectors $x_1,\ldots,x_m$ are introduced sequentially and repeatedly. At step t a vector x_k is presented and C_k in (2) stands for the cost function related to it. The process is iterated until convergence. At each time the weights are updated according to the rule:

$$w_{jh}(t) = w_{jh}(t-1) - \mu(t)\, \partial C_k / \partial w_{jh} \qquad (2)$$

where $w_{jh}(t)$ and $w_{jh}(t-1)$ are the weights from the neuron h to the neuron j at steps t and t-1 respectively when the pattern x_k is presented, and $\mu(t)$ is the gradient step at step t.

The GBP algorithm proceeds as follows :

1- At step t, an input pattern x_k is presented on the input layer.

2- The states of the automata are propagated forward up to the output layer, by using equation (1)

3- The computed and desired output s_k and y_k are compared. Then the cost error for the output layer is derived.

4- The weights are modified according to equation (2) and then one returns to step 1.

The process is stopped when the cost function C is smaller than a given chosen value

The following example (Fig 4) shows the weights computed by the GBP algorithm in order to realize a XOR (exclusive OR) function (XOR(1,1) = 0, XOR(0,0) = 0, XOR(1,0) = 1, XOR(0,1) = 1) with a 3-layer network.

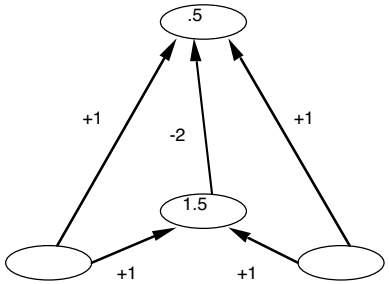


Figure 4: A network solution to the XOR function. In this example threshold automata is used corresponding to the case T= 0 in the sigmoid function; f is thus the step function with thresholds (As an example the threshold of the automaton of the intermediate layer is 1.5 and s_i =1 if $A_i \geq 1.5$ and 0 otherwise).

At the end of the learning phase, the "machine" is completely determined. Information or "knowledge" is encoded into the connections and the network is able to generalize, i.e. to associate new sets of data which have not yet been learned. The learning phase may lead to long computations due to the minimization process. But during the operational phase the computation time is very fast

because all the minimizations have been done during the learning phase and the only computations are algebraic operations.

1.4 Properties of the model

The general problem in supervised learning when dealing with neural networks is to realize an association between inputs and desired responses, i.e a mapping F which is of the form: (inputs) --> (desired responses). If this problem is solved by a quasi-linear multi-layered network, F has to be decomposed into successive non-linear functions like those presented in (1). Theoretical results [Funahashi, 1989] proved that by restricting F to be continuous, such a decomposition exist. The main theorem leads to the approximation of continuous mappings by l-layer networks. It is shown that any mapping whose components are summable on a compact subset, can be approximately represented by l-layer networks ($l \geq 3$) in the sense of L^2 norm. Such results come from a Kolmogorov theorem [Kolmogorov, 1957] which shows that any continuous mapping can be represented by a "form" of a four-layer neural network using a family of real functions. Unfortunately, the original framework does not state how to compute the functions needed by the particular decomposition. Experimental results show that quasi -linear functions provide a powerful basis of functions and that the theoretical results can often be extended to non-continuous mapping [Sejnowski, 1987].

In the present study Q.M.N is used as a classification system: the desired answer y_k is now the vector indexing the class ω_i of the input x_k. The problem is to classify the input patterns x_k in p different classes. The response y_k is of the form: $y_k = (y_{1k} ,...,y_{pk})$ with $y_{ik} = +1$ if x_k belongs to class ω_i and $y_{jk} = 0$ for $j \neq i$. We name F the set of functions represented by the architecture of the Q.M.N chosen in order to make the association. Theoretical results [Duda, 1973, White, 1989] allow one to compare the optimal mapping computed by the retro-propagation algorithm and the Bayes classifier. The optimal mapping gives the "best" approximation in F of the Bayes discriminant

function. By approximating this function, the computed output s_{ik} (i = 1....m) gives direct information about the a-posteriori probabilities $P(\omega_i / x_k)$ where ω_i stands for the p different classes.

Since the algorithm is adaptive and requires an initialization, knowledge of first guess given for w_{ij} is crucial in order to improve the speed of convergence. Thus we expect the work done on academic learning sets gives a valid approximation for the first guess when dealing with the real scatterometric wind fields and will reduce the computation time during the learning phase.

2. PRESENTATION OF THE METHOD

The method can be divided into two phases.

In a first phase, which is the learning phase, information is extracted from a learning set. An automata network able to remove ambiguities is built by comparing ambiguous wind field vectors measured by a scatterometer to real wind vectors determined by an "expert". In a second phase, which is the operational phase, the computed network is tested on new data : it is applied to scatterometer wind fields to remove directional ambiguities.

2.1 The Data-set

The method has been tested on winds obtained from the output of meteorological operational models. Satellite trajectory and scatterometer swath have been computed. Wind scatterometer data have then been sampled using wind fields provided by the ECMWF. The resolution of the ECMWF model is about 110km. Winds have been interpolated onto a grid of 50km*50km in order to obtain a realistic value of the resolution of scatterometers.

This data set is very regular with respect to the actual wind. In order to reproduce errors due to electronics and to simulate the space variability of the wind, a Gaussian noise has been added to the azimuth of each vector with a 10° rms and mean zero. This seems realistic according to Schultz (1990). This wind is called the actual simulated wind.

Inversion of the geophysical model function yields ambiguities onto the azimuth of the wind vector. In this study which can be mainly considered as a feasibility of the method, four likelihood estimates for wind directions have been simulated: the true direction, the 180° ambiguity and two ambiguities at +/- 90°. This ambiguity simulation is academic but not too far from the simulation of ERS.1 scatterometer (A. Long, 1985) since one antenna azimuth is perpendicular to the track and the other two are symmetric with respect to this antenna, making an angle of +/- 45°

The 180° ambiguities have been generated by randomly inverting a certain number of vectors of the simulated wind map. In order to introduce a rate of n% ambiguities, a number between [0,1] is chosen according to a uniform law. The direction is inverted if the number chosen is less than (0.**n**). In the first series of experiments **n**=30 and **n**=22.5 in the second. The 90° ambiguity is generated using the same procedure. Because this last ambiguity is less frequent than the 180° one, we chose **n**=7 in the first series and 2.5 after. The vectors of the simulated wind map are thus rotated randomly by an angle of +/- 90° .

The obtained wind field can be then considered as a rough simulation of the most likely wind field obtained from a wind retrieval procedure. The likelihood estimates for directions are ranked as

follows : the first one is supposed to be the most likely, the second is at 180° of the most likely, and the third and the fourth at +/- 90°. This simulated wind for a specific day is displayed in Fig. 5

One of the main advantages of automata network is that the approximations provided by the network are dependent on the real distribution of the data and are directly computed from the database. In particular, they do not require the Gaussian hypothesis. This method is very powerful in the sense that it can deal with any arbitrary noise

2.2 The QMN Architecture

It has been assumed that the wind observed on a specific cell is related to the wind observed on a certain neighborhood. So, in the first step the Q M.N method deals with local information. This topological relationship is modeled by a window of 5 * 5 elements (Fig. 6). The internal structure of the neural network associated with that window constitutes the core of the "machine" described in Section 2. This machine must be able to recognize most of the common patterns observed in meteorology, i.e. mid latitude cyclonic and anticyclonic motions, trade winds, and fronts; it must then remove ambiguities. Dealing with such a window, the information comes from these 25 measurements taken in a square box of 200 km *200 km. The size of the window must be large enough to obtain a significant probability to have enough wind vectors in the right direction in order to constrain the computed wind. It also must be small enough in order to be representative of the true wind without distortion due to space variability. The adequacy of the size of the window in relation to the efficiency of the ambiguity removal have been discussed by Schultz (1990) for the circular median filter. In the present work, experience has shown that there is enough information included in a 5*5 window to remove the ambiguities. This small window allows one to improve the speed of the

computation. Moreover the total process mixes local and global information and is able to cope with a rather small window.

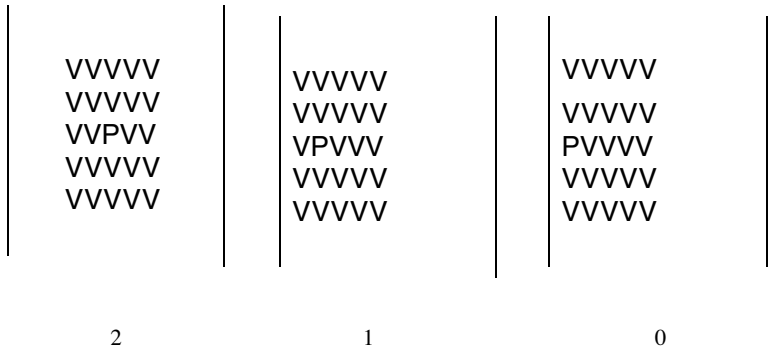


Figure 6: Presentation of the three classes of neighborhood for a 5*5 window. P is the cell to be examined, V are the neighborhood cells

The purpose of the "machine" is to determine if the wind vector at P is consistent with those observed at V (Fig.6). If not the wind direction at P will be changed.

The two ambiguities are solved separately. First we determine a machine (M1) to remove the 180° ambiguity; then another machine (M2) to remove the 90° ambiguity. The two machines are applied sequentially as in the relaxation processes used in image analysis. We first apply M1 on the swath studied then M2 and again M1 to remove the 180° ambiguities which could have been generated by the use of M2

The two machines are different in terms of weight only; their architecture and the method involved in the determination of their internal structure is identical. In the following we present the general method used for both of of them to determine the internal state of a machine.

Each "machine " has the same structure. It is made of three filters i.e. three multi-layered networks corresponding to the three windows described below. Considering the ratio of the window size to the swath width which is large, the side effect is an important problem to solve. In order to take it into account the side effects of the swath three windows have been considered, depending on the position of the studied point with respect to the side of the swath. Window 0 corresponds to a point located on the side, window 1 to a point located at 50 km from the side and window 2 to a point in the center of the window (see Fig 6). Each network therefore uses a specific learning set depending on the neighborhood taken into account, but has the same architecture

Each network is comprised of three layers :

The input layer is composed of 50 neurons: the first 25 are forced with the sine of the 25 wind vector azimuth of the considered window, the 25 others with the cosine of these azimuth. This code has been chosen in order to avoid a discontinuity at $2*\pi$, which would appear when dealing with the modulus and the azimuth of the vectors.

The single layer of hidden cells is composed of 25 neurons. This number which has been chosen from empirical considerations obtained from similar problems allows us to get good results during the learning phase and the working phase. We have not tried to obtain a minimal architecture; this will be done in the future and will allow us to decrease the number of calculations in the learning phase as well as in the working phase.

The Q.M.N is used in a classification mode as described before, so the output layer has two cells only, each of them corresponding to the expected answer: the azimuth is true, or the azimuth is false. The output values obtained on this last layer are related to a-posteriori probability and can be interpreted as a likelihood estimate of the accuracy of the direction: the greatest output value allows one to make the decision of inverting or not inverting the direction.

In the present case, two adjacent layers are fully connected (see Fig 7).

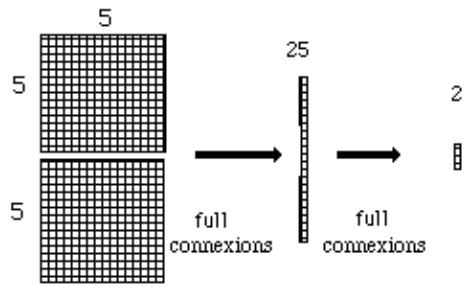


Figure 7 : Architecture of the network showing the input layer (on the left) corresponding to the sine and cosine of the wind vectors , the hidden layers (in the middle), and the output layer (on the right). Each cell of the input layer is connected to each cell of the hidden layer and each cell of the hidden layer is connected to the output layer

Because each point of the swath is related to one network only, the computation time necessary to remove the wind-ambiguities is proportional to the number of wind-measurements and to the window size. This time is very fast since the only computations are successive elementary algebraic computations. The minimization procedures which are time consuming have been done during the learning phase.

2.3 The Learning phase

As neural networks belong to supervised learning, the network has been trained to automatically remove ambiguities by memorizing a certain number of specific examples. A set of data has been chosen which was part of the eight simulated scatterometer swaths obtained during a specific day over a specific region (Fig.5). We have tried to take into account the different patterns which are present in the wind field. Thus, in this data set each specific meteorological situation appeared with the same frequency.

For each ambiguity, special test maps have been built where only the studied ambiguity is simulated on the actual simulated wind. The data sets used in the learning phase can be classified

into three types which are related to the three types of windows. Each data set has about 2000 elements. Because we are dealing with two type of ambiguities, six learning data sets have been considered, one for each network. For the 180° ambiguities the error rate generated is set at **n** = 30% and for the second at n = 7% on each map. Each network was then trained according to its learning set. The accuracy of each network has been tested separately, so we have computed the learning rate of each network related to its learning set.

2.4 The Generalization phase

Thereafter the two machines (i.e. each of the six networks) have been tested on new data sets in order to check the machine's ability to solve different meteorological situations and the possibility to use them on a general basis.

First, test maps dedicated to each ambiguity have been generated with the same rate of ambiguities and each network has been tested separately in order to evaluate its capabilities to dealiase its wind ambiguity. The dynamics of the process was then checked by applying the two "machines" M1 and M2 successively on test wind maps with both ambiguities.

3 THE RESULTS.

3.1 The learning phase

The learning phase was performed on a totality of 2000 examples, including three types of windows and for each ambiguity respectively. The data set was obtained from ECMWF wind simulation on August 18, 1985 in the South Atlantic Ocean. Table 1 shows the percentage of patterns correctly learned for each network related to each machine. These results have been obtained after 150 runs on the relative learning data set. At that time it was decided that enough information had been stored in the network.

Neighborhood	180°-ambiguity	90°-ambiguity
type 2	99%	100%
type 1	99%	98%
type 0	98,8%	99,5%

Table 1: Learning rate: rate of ambiguity retrieval over the learning set. (Memory is built with 2000 windows acting on August 18 wind map).

Note: no error was introduced by the "machine".

3.2. Generalization

The validity of the results concerning the windows of types 2 , 1 and 0 has been tested using the map of August 18 and on maps of three other days (August, 6, 16 and September 1) where the wind is supposed to be decorrelated from the wind observed August 18. In order to obtain realistic wind fields, noise has been added to the wind vectors in the same manner as for the learning data set (i.e. Gaussian noise with a rms of 10° and a mean equal to zero). In order to have a clear understanding of the performances of the method different tests have been performed.

In the first series of experiments only one ambiguity is removed during each test (30% of the vectors have been rotated by 180° on each map to test M1 and 7% by +/- 90° to test M2). The results are presented in tables 2 and 3.

In the second series both ambiguities where generated on the same map of August 16 (25% of the vectors were perturbed : 22,5% rotated by 180° and 2,5% by +/- 90°). The global process and the results are reported in section 3.3.

Neighborhood type	August 18	August 16	August 6	September 1
type 2	98%	97,5%	98%	98%
type 1	99.47%	98.5 %	99,12%	98,3%
type 0	98.93 %	98.2 %	99%	98,14%

Table 2: Generalization rate for the first ambiguity (180°) removal
(rate of pattern retrieval over the different maps)

Neighborhood type	August 18	August 16	August 6	September 1
type 2	99.95%	99.95%	99.95%	99.95%
type 1	97.86%	97.7 %	97.6%	97.23%
type 0	99.5 %	99.37 %	99.6%	98,96%

Table 3: Generalization rate for the second ambiguity ($\pm$90°) removal
(rate of pattern retrieval over the different maps)

3.3. Operational ambiguity removal process

As we have seen before the ambiguity removal process deals with both ambiguities at the same time using a relaxation process. The removal of each ambiguity is done with a "machine". A machine is composed of three neural networks adapted to deal with windows of types 0, 1 and 2, depending upon the position of the cell considered with respect to the swath. Two different machines are used: M1 to remove the 180° ambiguity and M2 to remove the 90° ambiguity. When using a machine, an output is computed for each wind vector within the swath. The largest output (which corresponds to the largest a-posteriori probability for the observed direction correct or incorrect) indicates the response.(yes or not). All responses are recorded in a buffer. In order to avoid possible propagations of errors, changes of directions only occur at the end of the computation from the information within the buffer. The maximum likely 180° ambiguity is easy to remove since one only has to change the azimuth of the wind. The removal of the 90° ambiguity implies a knowledge of the error's sign (+/- 90°). When a 90° ambiguity is detected, the azimuth of the wind vector is systematically rotated by +90°. A 180° error might have been generated by this procedure. However such errors will be easily removed by applying M1 a second time. The ambiguity removal process can be decomposed in the following manner:

We start from map n°1 where all ambiguities are present.

Stage 1: M1 is applied to map n°1 in order to remove the 180° ambiguities. A map n°2 is then obtained where most of the 180° ambiguities have been removed.

Stage 2: M2 is applied to map n°2. Most of the 90° ambiguities are removed but some 180° ambiguities might have been generated according to the positive rotation which has been applied to remove the 90° ambiguity. A map n°3 is then obtained.

Stage 3: M1 is applied to map n°3. M1 removes the 180° ambiguities which may not have been removed during the first passage because of the presence of the 90° ambiguities. M1 also removes the 180° ambiguities generated during the stage 2. A map n°4 is then obtained where most of the ambiguities have been removed.

The map n°4 corresponds to the maximum likelihood estimate of the wind.

The relaxation process could be run again in order to improve the performances. With simulated data it is noted that 3 stages provide sufficiently good performances: Figure.8 displays the maps obtained after the successive use of M1, M2 and then M1 again, in a region where the wind is highly variable. A total of 1387 ambiguities have been generated on the first map, and after the third passage a total of 1292 have been corrected. Thus the rate of exact directions at the end of the process is about 7900/8000, i.e. 0.9875 (or 98.75%).

CONCLUSION

The results presented above have proved the efficiency of neural networks in removing simulated wind ambiguities. A large variety of meteorological information is encoded in the machine, enough at least to solve the most common features which are encountered in the wind fields studied. This work is a first attempt dealing with simulated data in order to show the feasibility of the method. The neural networks have been trained, i.e. the weights w_{ij} of the connections of the neurons have been computed on a data set corresponding to one observation day only in the area studied. The machine built is then able to remove wind ambiguities in wind fields observed during other days, either lagging or leading the training day for several days during which wind fields are "a-priori " uncorrelated. This shows the ability of the method to be used on an operational basis.

This machine can be viewed as a generalization of space filters which are used in image processing. In the present case, the method determines the maximum likelihood estimates of the wind direction to be correct by computing the weights w_{ij} of the so-called filter. In such methods, the size of the window is a crucial parameter: the window size must be large enough to constrain the ambiguity removal process by topological relationship and small enough to take into account the space variability of the wind. Dealing with a circular median filter Schultz (1990) found that a window size equal to 7 is a good compromise. In the present problem a window size equal to 5 gives very consistent results which proves the efficiency of the method. Furthermore, the smaller the size, the quicker the computation; there exists a factor 2 between computation done with a window size equal to 5 with respect to a window size equal to 7.

The relaxation process used prevents error propagations. The response given by the neural network during the scanning of the studied wind field are recorded in a buffer. Changes in direction only occur at the end of the computation from the information within the buffer without any possible rejection of one change over another. Due to this procedure and to the overlapping of the different windows, a stage of the ambiguity removal algorithm deals with the quasi-totality of the information included in the map and not only with that located in the vicinity of the window. So the total process mixing local and global information is able to cope with relatively small windows.

With real data we would have proceeded in the same manner. At each wind vector, the geophysical model function associates several azimuths χ_k (k = 1,.....p) ranked according to their probability. M1 instead of making a 180° turn when the vector is designed to be ambiguous, would change χ_1 into χ_2 if the response is positive.. Then a second scanning with M2 would change χ_2 into χ_3 if the response is positive.

Several important aspects occurring in ambiguity removal have not been studied as clusters and spatially correlated errors. These errors can dramatically affect the skill of the method. The main objective of this paper is not to propose an operational scheme for dealiasing scatterometric wind field but rather to draw attention to the performances of a new method which has never been used in oceanography or in meteorology before. In fact, the wind ambiguity removal algorithm presented

here is one element of a general scatterometer wind retrieval procedure we are presently working on. The complete chain will be presented in a forthcoming paper where a large variety of realistic cases are solved.

In conclusion we would like to stress that neural network techniques present an interesting new approach to the scatterometric wind ambiguity removal. Its skill is comparable to other methods as the circular median filter of Schultz (1990) which presents some similarities with our method as the use of window and the implicit hypothesis that enough information is encoded on a specific neighborhood to remove ambiguities. Its major advantages are linked to the rapidity of computation since neural networks can easily be programmed on parallel computers (Rumehart and Macclelland, 1986). Hence, the machine is well adapted to solve problems in real time.

As in any method used in geophysics, we will have to face the complexity of real data when the scatterometers start flying and giving actual measurements. Due to the statistical properties associated with neural networks, which compute approximations of Bayesian probabilities without any extra assumptions like the Gaussian Hypothesis, we expect to get similar results. By that time, the major problem will be to create an exact learning set with correct wind fields associated with the scatterometer measurements since the final performances of the algorithm are based on the representativity of the training set.

Acknowledgements

This work was supported by the French Institutions : Centre National de la Recherche Scientifique and Programme National de Télédétection Spatiale. We are grateful to Dr V. Cassé for useful discussions. We thanks P. Braconnot and M. Boukthir for providing the data set.

BIBLIOGRAPHY

AARTS E. and J.KORST: *Simulated annaeling and Boltzmann machines* Wiley, 1989

AZENCOTT R.: *General Boltzmann machines with multiple interactions* . IEEE PAMI 90, to appear

[

BOURLARD H. and J.C. WELLEKENS: *Speech, pattern discrimination and multilayers perceptrons*. Computer, Speech and Language, vol 3 pp1-19, 1989.

[

BADRAN F, S. THIRIA and M. CREPON: *A method to de-alias the scatterometer wind field: a real world application*. NATO ARW, Neuro Computing, Algorithms, Architectures and Applications. Les Arcs, 1989.

CAVANIE A. and D. OFFILER: *ERS-1 wind scatterometer: wind extraction and ambiguity removal*. Proc. of IGARSS 86, ESA SP-254, vol. 1, pp. 395, 398, 1986.

DEVIJVER P. and J. KITTLER: *Statistical Pattern Recognition*. PRENTICE HALL, 1982.

DUDA R.O. and P.E. HART: *Pattern Classification and Scene Analysis*. WILEY, 1973.

FREILICH M. H. and D.B. CHELTON: *Wave number spectra of Pacific winds measured by the Seasat scatterometer,* J. Physc. Oceanogr, 16, 741-757, 1986.

FUNAHASHI K. I. : *On The Approximate Realization of Continuous Mappings by Neural Networks*. Neural Networks, vol. 1, pp. 183-192, 1989.

FOGELMAN SOULIE F., P. GALLINARI, Y. LE CUN and S. THIRIA: *Automata networks and Artificial Intelligence*. In "Automata networks in computer Science", Manchester Univ. Press, Princeton Univ. Press, 1987.

HINTON G.E.: *Connectionist learning procedures*. Technical report CMU-CS 87-115 Carnegie Mellon University, 1987

HOFFMAN, R. N.: *SASS wind ambiguity removal by direct minimization*. Mon. Weather Rev., 110, 2197-2206, 1982.

HOFFMAN, R. N.: *SASS wind ambiguity removal by direct minimization, 2, Use of smoothness and dynamical constraints*. Mon. Weather Rev., 112, 1829-1852, 1984.

KOHONEN T. : *Self organization and associative memory*. Springer Series in Information Sciences, vol 8, Springer Verlag, second edition, 1987.

LIPPMANN R.P. : *An introduction to computing with neural nets*. IEEE ASSP Magazine, pp. 4-21, 1987.

LONG A.E. : *Toward a C-band radar sea echo model for the ERS-1 scatterometer*. Proc. 3rd coll. on Spectral Signatures. ESA SP - 247, pp. 29-34, 1985.

PRESS W.H, B. FLANNERY, S. TEUKOLSKY, W. VETTERLING: *Numerical recipes in C*. Cambridge University press, 1988

ROQUET H and A. RATIER: *Toward direct variational assimilation of scatterometer backscatter measurements into numerical weather prediction model*. Proc. of IGARSS 88 Symposium. Ref ESA SP-284, pp. 257-260, 1988

RUMELHART D.E.and J.L. MACCLELLAND: *Parallel distributed processing: explorations in the microstructures of cognition*, MIT Press, 1986.

SEJNOWSKI T.J. and C.R. ROSENBERG : *Parallel networks that learn to pronounce English text*, Complex Systems, 1, pp. 145-148, 1987.

SCHWENZFEGER K.J. : *Algorithm for wind scatterometer data analysis ambiguity suppression*. Proc. Conf. On Use of Sat. Data in Climate Models. ESA - SP 244, 1985.

SCHULTZ, H: *A median filter approach to correction errors in a vector field*. IEEE Proc. of the IGARRS 85 Symposium, 1985.

SCHULTZ H. : *A circular median filter approach for resolving directional ambiguities in wind fields retrieved from spaceborn scatterometer data*.. J. Geophys. Res. , 95, 5291-5304, 1990.

TAMURA S.and A. WAIBEL: *Noise Reduction Using Connectionist Models*. ICASSP, pp. 553-556, 1988.

WIDROW B. and M.E HOFF: *Adaptive switching circuits*. IREWESCON conv, Record part 4, 96-104, 1960.

WIDROW B. and S.STEANS: *Adaptive signal processing*. Prentice Hall signal processing series, Alan V Oppenheim series editor, 1985.

WHITE, H: *learning in Artificial Neural networks: a statistical perspective neural computation*, vol 1 n°4, 1989.

Figure captions

Figure 1: (a) Diagram of an automata (neuron). (b) Sigmoid function f for different values of α . The figure shows different functions f for T varying from 4 to 1 (the smallest T , the steepest the curve)

Figure 2 : Multi-layered network

x_k represents the input vector, s_k the output computed by the network and y_k the actual output to which s_k must converge

Figure3: Associative memory.

An input x_k is presented to a machine M (the Multi-layered-Network). The transfer function F associated with M allows the computation of an output y_k associated with the input x_k

Figure 4: A network solution to the XOR function. In this example threshold automata is used corresponding to the case T= 0 in the sigmoid function; f is thus the step function with thresholds (As an example the threshold of the automaton of the intermediate layer is 1.5 and s_i =1 if $A_i \geq 1.5$ and 0 otherwise).

Fig 5 Simulated scatterometer wind field obtained from output of ECMWF on march the 18th in the South Atlantic ocean.

Figure 6: Presentation of the three classes of neighborhood for a 5*5 window. P is the cell to be examined, V are the neighborhood cells

Figure 7 : Architecture of the network showing the input layer (on the left) corresponding to the sine and cosine of the wind vectors , the hidden layers (in the middle), and the output layer (on the right). Each cell of the input layer is connected to each cell of the hidden layer and each cell of the hidden layer is connected to the output layer

Fig. 8 Different stages of the ambiguous wind field during the removal process: Map 1 corresponds to the initial wind field where the ambiguous vectors are present and characterized by a small circle. A Gaussian noise has been added to the azimuth of each vector with a 10°rms in order to simulate noise due to electronics and space variability of the wind; 30% of the wind directions.are ambiguous b) Map 2 displays the wind field after the use of M1 which removes the 180° ambiguities. c) Map 3 displays the wind field after the use of M2 which removes the 90° ambiguities. d) Map 4 display the wind field after a second passage of M1 in order to remove the 180° ambiguities which might have been generated by the use of M2. The map 4 wind field corresponds to the most likelihood estimate of the wind.

Neighborhood	180°-ambiguity	90°-ambiguity
type 2	99%	100%
type 1	99%	98%
type 0	98,8%	99,5%

Table 1: Learning rate: rate of ambiguity retrieval over the learning set.
(Memory is built with 2000 windows acting on August 18 wind map).
Note: no error was introduced by the "machine".

Neighborhood type	August 18	August 16	August 6	September 1
type 2	98%	97,5%	98%	98%
type 1	99.47%	98.5 %	99,12%	98,3%
type 0	98.93 %	98.2 %	99%	98,14%

Table 2: Generalization rate for the first ambiguity (180°) removal
(rate of pattern retrieval over the different maps)

Neighborhood type	August 18	August 16	August 6	September 1
type 2	99.95%	99.95%	99.95%	99.95%
type 1	97.86%	97.7 %	97.6%	97.23%
type 0	99.5 %	99.37 %	99.6%	98,96%

Table 3: Generalization rate for the second ambiguity ($\pm$90°) removal

(rate of pattern retrieval over the different maps)