A Hopfield neural network to track drifting buoys in the ocean
Parisi, V.1, Garcia, E.2, Cabestany, J.1, Font, J.2, Salas, J.2
1 Dept. Enginyeria Electronica, Polytechnical University of Catalonia, EUETIT, c/ Colom, 1, 08222 Terrassa (Spain), parisi@eel.upc.es
2 Dept. de Geologia Marina i Oceanografia Fisica, Institut de CiÉncies del Mar, (C.S.I.C.), 08039 Barcelona (Spain)

Abstract
A methodology is proposed to estimate sea surface currents from the information given by the sea surface temperature (SST) obtained from satellite images. Currents are estimated from the motion field of a temporal sequence of images using a Hopfield neural network. A cost function is minimized when some rules of correspondence between pixels in successive images are considered. A performance test of our methodology is done using the information from the paths made by lagrangian buoys. The simulations, done using the buoys positions, prevent us to use a system able to escape from the local minima, where the Hopfield neural network uses to get trapped, until reaching a global minima.

I. Introduction
Remote sensing actually becomes one of the most important techniques to get information about the sea state. Data of the sea surface temperature (SST), the ocean surface color and the surface rugosity of the sea surface elevation are important parameters to understand the ocean and atmospheric dynamics and now they are available from remote sensing; The interest of such information lies in the synopticity of the data and also in the great temporal and spatial resolution. However, problems appear when we want to extract dynamical information of remote sensing observations. This problem becomes serious if one is interested in the dynamics of mesoscale structures. They are fast evolving features with complex spatial patterns. Image processing techniques are however promising tools to obtain more quantitative analysis in a quasi-automatic way.

Our particular interest is focused on estimating sea surface currents from AVHRR and color images. In such case, this estimation may be done by computing the apparent motion between consecutive images. When large displacements occur and the temporal sampling period is high an approach to compute it consist in finding the correspondence between structures identifiables, in successive images then to track their positions.

Based on this approach Côte et al. (1995, 1997) use a Hopfield neural network to match points in pairs of consecutive images. They found that it performs better than the more classical correlation methods (Emery et al. 1986) specially in areas with strong local variations of current direction and magnitude. In any case, the performance of such estimators has to be done by comparing with those obtained using another techniques on the same data.

Here we propose a methodology based on the ideas of Côte et al. and present the results obtained when applied to a test case where the trajectories of drifting buoys are already known. The points describing such trajectories represent pixels of an structure in consecutive images.

The paper is organized as follows: section II is a brief introduction to the Hopfield neural network and its use to solve combinatorial optimization problems. In section III gives the main ideas of Côte et al. Approach. In sections IV and V we describe in detail our proposal and we show the results of our simulations. Finally we conclude and briefly discuss future improvements that can be done.

II. The Hopfield neural network
The model of neural network proposed by Hopfield (1982) consists in a set of fully interconnected discrete neurons, with two possible output states \(v_i^0\) unactive, \(v_i^1\) active. Each one of the neurons receives signals \(I_i\) from external sources and signals \(v_j\) from the rest of neurons of the network, as depicted in Fig. 2.1, weighted by
a synaptic interconnection $T_{i,j}$ as states the expression (2.1).

Input to the neuron $u_i$ is the sum of both terms.

![Neuron diagram showing its inputs and output connections.](image)

**Fig. 2.1.** Neuron diagram showing its inputs and output connections.

$$u_i = \sum_{j=1}^{M} T_{i,j} v_j + I_i$$  \hspace{1cm} (2.1)

Neurons are selected randomly to update their output, that becomes active if $u_i$ is greater than a threshold activation value, $U_i^{th}$, or unactive if $u_i$ is lower than $U_i^{th}$

$$v_i \rightarrow v_i^0 \text{ if } v_i > U_i^{th}$$
$$v_i \rightarrow v_i^1 \text{ if } v_i < U_i^{th}$$  \hspace{1cm} (2.2)

The state of the network is defined as the set of binary outputs of all the neurons.

If the synaptic interconnections are symetric ($T_{i,j} = T_{j,i}$) and there are no self feedback connections ($T_{ii} = 0$) it's possible to define an energy function $E$ for the network,

$$E = -\frac{1}{2} \sum_i \sum_j T_{i,j} v_i v_j - \sum_i I_i v_i + \sum_i u_i U_i^{th}$$  \hspace{1cm} (2.3)

The change $\Delta E$ in energy whenever there is a change $\Delta v_i$ in a neuron is given by

$$\Delta E = -\left[ \sum_j T_{i,j} v_j + I_i - U_i^{th} \right] \Delta v_i$$  \hspace{1cm} (2.4)

In this expression, $\Delta v_i$ is positive when the terms in brackets is positive, defined as the updating condition in (2.2), and $\Delta v_i$ is negative when the term in brackets is negative. So the energy is decreasing at every neuron's change.


The ability to minimize an energy function in a very short convergence time makes it very useful to solve those problems whose solution can be found by minimizing a cost function.

The cost function is rewritten in the form of the energy function in order to find the synaptic weights $T_{i,j}$ and the external input $I_i$. This network evolves from a random initial state until it reaches a stable state, defined as the solution of the problem, where the energy has a minimum.

As an example, based on the ideas from Côte et al. (1997), imagine an optimization problem were $M$ elements of a group $A$ have to be identified in a group $B$, of $M$ elements too. Each one of the elements in $A$, $(A_i)$, has some features describing it that can be used to measure its similarity to the elements in group $B$, $(B_j)$. This similarity is formulated as a function $m(A_i, B_j)$, being low when there is a high similarity between $A_i$ and $B_j$ and high when they are very different.

A network of neurons is organized in a matrix structure, like that of fig. 2.2 below, where the columns represent elements of $B$ and the rows elements of $A$. Activation of a neuron $n_{i,j}$ represents a matching between elements $A_i$ and $B_j$. 
The cost function to be minimized with this network is

\[ C = w_m \left( \sum_i \sum_j y_{i,j} m(a_i, b_j) \right)^2 + w_c \left( \sum_i \left( \sum_j y_{i,j} - 1 \right)^2 + \sum_j \left( \sum_i y_{i,j} - 1 \right)^2 \right) \]  

(2.5)

The first term in this cost function (2.5) is minimum when the active neurons \((v_{i,j} = 1)\) are those representing similar elements \((m \text{ low})\). The second term express the constraint that there can be just one element in \(B\) matching an element in \(A\), so in each row and each column just one neuron can be active. When this occurs this term is null.

### III. Tracking methods of Côte and Chang

Côte et al. (1995, 1997) find 'interesting points' in pairs of consecutive AVHRR images and solve its correspondence through matching. The displacement of these interesting points is an estimation of the sea surface current.

The 'interesting points' are those points located on temperature contours of high curvature and where the temperature changes rapidly. The features used to match points are the curvature of the contour \(r_i\), the angle of orientation of local curvature \(\theta_i\), and its position \((x_i, y_i)\).

The energy (cost) function that represents the problem has two parts: the objective and the constraints.

\[ E = \text{objective} + \text{constraints} \]  

(3.1)

In the objective, the similarity between features of points in successive images is measured by a function \(\text{diff}(i, j)\). The similar the features are the lower is this function:

\[ \text{diff}(i, j) = w_1 \max \left( \max \left( \frac{r_i - r_j}{r_i - R^{*}} \right) - R^{*} \theta + \theta \right) + \]

\[ + w_2 \max \left( \min \left( \frac{\theta_i - \theta_j}{2\pi} \right) - \theta_i \right) - \theta \right) + \]

\[ + w_3 \max \left( \text{dist}_{ij} - \text{dist}_{i,j}^{*} \right) \]  

(3.2)

dist\(_{ij}\): distance between points \(i\) and \(j\)

The constraints, penalize that solutions that indicate different direction and magnitude for the current in a small neighborhood. Chang et al. (1993) find trajectories of objects from their sampled positions. They assume that motion of an object is smooth at any instant of time to find which are the most probable trajectories. This is done using a Hopfield neural network that minimizes an energy function where the direction and speed continuity of possible trajectories is taken into account. The likelihood of each possible is quantified by a path coherence function that measures it’s change in direction and speed. As an example, in the trajectory depicted in fig. 3.1 vectors \(\vec{v}_{i-1,i}\) and \(\vec{v}_{i+1,i}\) show the path followed by an object, from the sampling instant \(i-1\) until instant \(i+1\), where a change in speed and direction takes place.

Fig. 3.1: Vectors showing path direction in a possible trajectory
At instant, $i$, the path coherence is

$$\phi(\vec{v}_{i-1,j}, \vec{v}_{i,j+1}) = w_1 \left(1 - \frac{\vec{v}_{i-1,j} \cdot \vec{v}_{i,j+1}}{||\vec{v}_{i-1,j}|| \cdot ||\vec{v}_{i,j+1}||}\right) + w_2 \left(1 - \frac{2||\vec{v}_{i-1,j}||^2}{||\vec{v}_{i-1,j}||^2 + ||\vec{v}_{i,j+1}||^2}\right)$$

(3.3)

And the coherence of all the trajectory is the sum of the coherences at each sampling instant.

IV. Estimation of sea surface currents in a sequence of satellite images

The method is proposed to find the correspondence in successive images of tokens, found in each image, using as a similarity measure, the distance between points, the difference in their features, and the speed and direction coherence of the trajectories.

It makes a sequential analysis using three images in each step, as shown in fig. 4.1.

Suppose that the points $p_{1,i}, p_{2,j}, p_{3,k}$ ( $p_{1,m}$: interesting point number $m$ found in the $l_{th}$ image) are in the same trajectory; to measure the speed and direction coherence two vectors $\vec{r}_{1,i;2,j}$ and $\vec{r}_{2,j;3,k}$, linking these points are defined, as shown in the following graph.

![Figure 4.2: Vectors showing the path of possible trajectories](image)

The measure of likelihood of this trajectory is expressed as

$$l(p_{1,i}, p_{2,j}, p_{3,k}) = w_l \max \left\{1 - \frac{\vec{r}_{1,i;2,j} \cdot \vec{r}_{2,j;3,k}}{||\vec{r}_{1,i;2,j}|| \cdot ||\vec{r}_{2,j;3,k}||}\right\} - l_a \max, 0\right) + w_s \max \left\{1 - \frac{2||\vec{r}_{1,i;2,j}||^2}{||\vec{r}_{1,i;2,j}||^2 + ||\vec{r}_{2,j;3,k}||^2}\right\} - l_s \max, 0\right) + w_m m(p_{1,i}, p_{2,j}) + w_m m(p_{2,j}, p_{3,k})$$

(4.1)

Function $l(p_{1,i}, p_{2,j}, p_{3,k})$ has very low values for likely trajectories and gets very high when the path between points $p_{1,i}, p_{2,j}, p_{3,k}$ could rarely be assigned to a trajectory. The first and the second term are those functions given by Chang et al. to measure direction and speed coherence, respectively. Trajectories with higher coherence than a threshold value, expressed by $l_a \max$ and $l_s \max$, give null values to these terms. The third and fourth term account for the distance and similarity between the pairs $p_{1,i}, p_{2,j}$, and $p_{2,j}, p_{3,k}$.
First term measures the distance \( d(c_{i,k}, c_{i+1,j}) \) between points \( p_{i,k} \) and \( p_{i+1,j} \). The rest of the terms measure the differences in the rest of available features. Each term is weighted independently and there is a threshold for each one; distances or differences lower than this threshold represent identical features, so the term is null.

A Hopfield neural network is arranged in a three dimensional grid, as a cube, as shown in fig. 4.3 below. Each neuron is associated to one of the possible trajectories: neuron \( n_{i,j,k} \) is associated to the trajectory linking points \( p_{i,j}, p_{2,j}, \) and \( p_{3,k} \). This network minimizes a cost function that reaches a minimum when the trajectories selected are the most likely.

\[
m(p_{i,k}, p_{i+1,j}) = w_{md} \max(d(c_{i,k}, c_{i+1,j}) - d_{\min}, 0) + \\
+ w_{mf1} \max\left| f_{i,k,1} - f_{i+1,j,1} \right| - \Delta f_{1\min}, 0) + \\
+ w_{mf2} \max\left| f_{i,k,2} - f_{i+1,j,2} \right| - \Delta f_{2\min}, 0) + \\
+ \cdots + \cdots + \cdots \\
+ w_{mfNf} \max\left| f_{i,k,N} - f_{i+1,j,N} \right| - \Delta f_{Nf\min}, 0) \\
\]

\( Nf \): Number of features

The first term sums the likelihood of the trajectories associated to those neurons active, the three next terms are zero if the points in the three images are associated with just one or zero trajectories.

Once the first three images are analyzed the results are points forming trajectories and points not assigned, and it proceeds sequentially with the rest of images.

The points each new image have to be assigned to a trajectory, to a point in its previous image or not assigned to any. To do this, each one of the new points is compared to all points in the last already analyzed image with the aid of the function \( m(l(p_{i,j}, p_{i-1,k}), p_{1,i}) \), were \( p_{i,j} \) is a point \( (j) \) in the new image \( (i) \), and \( p_{i-1,k} \) is a point \( (k) \) in the anterior image.

If the previous point belongs to a trajectory, the function determines how likely the new point is in the same trajectory. If the previous point doesn't belong to any trajectory, the function determines the matching between these two points. This function is expressed as

\[
m(l(p_{i,j}, p_{i-1,k}), p_{1,i}) = \\
\begin{cases} 
  \{l(p_{i,j}, p_{i-1,k})\} & \text{if } p_{i-1,k} \text{ belongs to a trajectory with } p_{i,j} \\
  \{m(p_{i,j}, p_{i-1,k})\} & \text{if } p_{i-1,k} \text{ doesn't belong to a trajectory} 
\end{cases}
\]

The expression of the cost function to minimize is
A Hopfield neural network is arranged as a matrix. Each neuron is associated to a point in the new image and one in the previous images: neuron \( n_{j,k} \) is associated to points \( p_{i,j} \) and \( p_{i-1,k} \). This network minimizes a cost function that reaches a minimum when the trajectories found, after assigning the new points are the most likely, with the constraints that one new point can be assigned to just one or zero previous points.

The expression of the cost function is:

\[
C = w_{	ext{cm}} \left( \sum_{j=1}^{N_p} \sum_{k=1}^{N_p_i} m(l_{i,j,k} - l_{i,j,k-1}) \right)^2 + \\
+ w_{	ext{cr}} \left( \sum_{j=1}^{N_p} \sum_{k=1}^{N_p_i} v_{j,k} - 1 \left( \sum_{k=1}^{N_p} v_{j,k} \right) \right) + w_{	ext{cc}} \left( \sum_{k=1}^{N_p_i} \sum_{j=1}^{N_p} v_{j,k} - 1 \left( \sum_{j=1}^{N_p} v_{j,k} \right) \right) \tag{4.5}
\]

V. Simulations and results

Real trajectories of a group of drifting buoys, released during ALGERS experiment, and whose local position is known approximately each 6 hours, are resampled each 24 hours to simulate the interesting points that would be found in SST images. These data serve as points to track and also to set some weighting coefficients.

Two features are added artificially to each sample, it’s curvature and an angle of orientation. In segments of linear trajectory this curvature is given by the curvature of a circumference, with a radius greater than 20 km. This value is chosen randomly for the initial sample of each trajectory and it’s changed along each sample in a range of 1%. In the case of circular trajectories, the curvature is chosen lower than 10 km, and the change within 10% of the initial value.

Orientation is kept nearly constant in linear trajectories and rotating, according to the circle described in circular trajectories.

Weighting coefficients of the likelihood function, measuring direction and speed coherence, are set using the variance of direction and speed of a group of buoys, not used to track. The higher the variance the lower are the coefficients. The rest of weights are set empirically.

VI. Conclusions

After an evaluation of the cost function used to solve the correspondence problem we have seen that it really reaches a global minimum at the right solution, but the presence of many local minima are an inconvenience because of the impossibility of the Hopfield neural network to escape from these stable states where it stacks. Thus an alternative system to reach the global minima, like the simulated annealing strategy, should be used.

Other questions appear, whilst increasing the number of features to be matched doesn’t increase the complexity of the neural network, it adds incognites to the system as by the moment the coefficients of the matching function are set empirically, and by the moment we can’t say if it will lead to a better performance.

A system able to reach a global minima of the cost function would help as to answer these questions.
Acknowledgments

We would like to express our gratitude to Dr. A.R.L. Tatnall and his group, at the Dept. of Aeronautics and Astronautics of the University of Southampton (U.K.), who received us and introduced to their methodology, where this work is based.

We also give thanks to the people of Addlink Software Cientifico S.A., who have kindly let us the Matlab package used to program and develop this work.

This work has been undertaken in the framework of the Mediterranean Targeted project (MTP phase II-MATER). We acknowledge the support from the European’s Commission’s Marine Science and Technology Programme (MAST III) under contract MAS-CT96-0051.

References


