Neural Networks for Improved Tracking

Leonid I. Perlovsky and Ross W. Deming

Abstract—In this letter, we have developed a neural network (NN) based upon modeling fields for improved object tracking. Models for ground moving target indicator (GMTI) tracks have been developed as well as neural architecture incorporating these models. The neural tracker overcomes combinatorial complexity of tracking in highly cluttered scenarios and results in about 20-dB (two orders of magnitude) improvement in signal-to-clutter ratio.

Index Terms—Combinatorial complexity, ground moving target indicator (GMTI) radar, multitarget tracking, neural networks (NNs).

I. INTRODUCTION

Ground moving target indication (GMTI) radar signals have been used for tracking since the 1990s [1]. When clutter is strong, so that signals are below clutter, detection and tracking are difficult. The Cramér–Rao bound for tracking [2] indicates that performance of the state-of-the-art algorithms is significantly below the information-theoretic limit.

The current GMTI detection and tracking subsystem operates in a two-step process. First, Doppler peaks are detected that exceed a predetermined threshold. Second, these potential target peaks are used to initiate tracks. This two-step procedure is a state-of-the-art approach which is currently used by most tracking systems. The limitation of this procedure is determined by the detection threshold. If the threshold is reduced, the number of detected peaks grows quickly. Increased computer power does not help because the processing requirements are combinatorial in terms of the number of peaks, so that a tenfold increase in the number of peaks results in a million fold increase in the required computer power.

The limiting factor for existing tracking algorithms is combinatorial complexity (CC) and not the absence of information in the data. CC has been encountered since the 1950s in various applications and was manifest in a variety of mathematical techniques. An early case was Bellman’s discussion of the “the curse of dimensionality” [3]. The following 30 years of developing self-learning mathematical techniques led to a conclusion that these approaches often encountered CC of learning requirements. The required examples had to account for all possible variations of “an object,” in all possible combinations with other objects, conditions, etc. The number of combinations quickly grows with the number of objects and conditions; say, a medium complexity problem of recognizing 100 different objects in various combinations might lead to a need to learn 100100 combinations; this number is larger than the number of all elementary particle interactions in the entire life of the Universe [4].

Initially, tracking algorithms did not face combinatorial complexity [5] and simple track models were used. Kalman filters revolutionized tracking [6] by providing the possibility to use complex track models. However, Kalman filters were designed for tracking single targets. For multiple targets in clutter, radial returns had to be associated either with clutter, or with a track, and, therefore, association algorithms had to be developed. Multiple hypothesis tracking (MHT) [7] is a widely used algorithm which evaluates multiple hypotheses about how the set of radial returns are associated with each track or with clutter. It is well known, however, to face CC [8]. To alleviate the CC of MHT, the probabilistic data association (PDA) algorithm was developed [9]. However, it can only maintain tracks—it cannot be used to initiate tracks in clutter.

Neural algorithms for tracking were developed based on Widrow’s adaline [10], a pioneering neural network (NN) combining neural architecture and a model-based structure suitable for tracking a single target. In this letter, a neural architecture is developed for tracking multiple targets in clutter, while avoiding combinatorial complexity. The tracker is tailored for GMTI data, and it follows the neural modeling fields architecture described in [8].

II. NEURAL MODELING FIELDS FOR GMTI TRACKING

For GMTI tracking, in neural modeling fields (NMF) theory [8], each input neuron \( n = 1, \ldots, N \), encodes four values: range and cross-range positions of cell \( r(x_n, y_n) \) and radial return parameters amplitude and Doppler \( A_n, D_n \); we denote \( \mathbf{X}(n) = (x_n, y_n, A_n, D_n) \). Every radial return is also characterized by time \( t_n \). We call an input neural field a set of \( \mathbf{X}(n) \); it is a set of bottom-up input signals. Top-down, or priming signals to these neurons, are generated by models, clutter and track models \( \mathcal{M}_u(S_h, n) \) enumerated by index \( h = 1, \ldots, H \). Each
model is characterized by its parameters $S_k$. In this letter, we consider targets moving with constant velocities $(v_{x_0}, v_{y_0})$

$$M_k(S_k, n) = (x_0 + v_{x_0} t_n, y_0 + v_{y_0} t_n, a_k, D_k).$$  \hspace{1cm} (1)

These models predict expected positions of targets. The NN compares these predictions to the data and generates parameter update signals as described later. Model parameters $S_k = (x_0, v_{x_0}, y_0, v_{y_0}, a_k, D_k)$ characterize position, velocity, average amplitude, and Doppler of the target, also $D_k = v_{x_0}$. These parameters are not known and are to be estimated through the internal dynamics of the NMF NN.

Interaction between bottom-up and top-down signals is determined by neural weights. First, define similarity measures between bottom-up signals $X(n)$ and top-down signals $M_k$

$$l(n|h) = \frac{1}{2\pi}\frac{1}{\text{det} C_h}\exp \left(-\frac{1}{2}(X(n) - M_k)^T C_h^{-1} (X(n) - M_k) \right).$$  \hspace{1cm} (2)

Here, we use a Gaussian function with the mean $M_k$, and a diagonal covariance $C_h$. Then the similarity measures are computed by modeling neurons, therefore, the similarity measure (2) was proven in [8]. Such local convergence usually occurs within relatively few iterations; a typical example in Section III took 20 iterations. Since similarity is a highly nonlinear function, regular convergence to the global maximum cannot be expected. The local rather than global convergence sometimes presents an irresolvable difficulty in many applications. In the presented method, this problem is resolved in three ways. First, the large initial standard deviation of the similarity measure smooths local maxima. Second, according to the previous description [before (10)], a large number of tracks is used initially; many of them are reinitiated several times. Therefore, if a particular real track is not “captured” after few iterations, it will be captured at a later iteration, after track reinitialization. Third, some of the initiated track models converge to spurious events not corresponding to real tracks; say, they will come nearby only two data points. In these cases, local LLR (10) will be low and spurious events are discarded. A detailed characterization of performance usually requires operating curves [12], plots of probability of detection versus probability of false alarm, computed for various signal-to-clutter ratios, densities of targets, target velocities, and other scenario parameters. Such detailed characterization is beyond the scope of this letter. We would just add that the NMF tracker performance came close to the Cramér–Rao bound for tracking [2] in several cases, when we performed such an investigation; these studies will be published separately. Finally, in practical applications, detecting a track is only a part of the overall tracking procedure. The detection procedure described in this section and illustrated in Section III is performed many times, as new data are acquired. Detected short-term tracks (tracklets) are connected into long-term tracks. In this process, spurious events are discarded, and tracks not detected initially are detected at a later time. This procedure, however, is not a subject of this letter. Graphical representation of the described NMF architecture is shown in Fig. 1.

$$f(h|n) = \sum_{h' \in H_k} r(h') l(n|h')$$ \hspace{1cm} (3)

Here, $c = \sigma x_0^2 / \sigma D_k^2$. For the unknown parameters $y_{0_0}$ and $v_{y_0}$, (7) is a linear system of equations; similarly, (8) is a 2-D linear system of equations for $x_{0_0}$ and $v_{x_0}$. After parameters are computed as previously described, standard deviations are computed as follows:

$$\sigma = \sigma_{\text{initial}} \exp \left(\frac{-\tilde{\gamma}}{TT}\right) + \sigma_{\text{final}}.$$  \hspace{1cm} (9)

Convergence of this iterative procedure to a local maximum of similarity measure (2) was proven in [8]. Such local convergence usually occurs within relatively few iterations; a typical example in Section III took 20 iterations. Since similarity is a highly nonlinear function, regular convergence to the global maximum cannot be expected. The local rather than global convergence sometimes presents an irresolvable difficulty in many applications. In the presented method, this problem is resolved in three ways. First, the large initial standard deviation of the similarity measure smooths local maxima. Second, according to the previous description [before (10)], a large number of tracks is used initially; many of them are reinitiated several times. Therefore, if a particular real track is not “captured” after few iterations, it will be captured at a later iteration, after track reinitialization. Third, some of the initiated track models converge to spurious events not corresponding to real tracks; say, they will come nearby only two data points. In these cases, local LLR (10) will be low and spurious events are discarded. A detailed characterization of performance usually requires operating curves [12], plots of probability of detection versus probability of false alarm, computed for various signal-to-clutter ratios, densities of targets, target velocities, and other scenario parameters. Such detailed characterization is beyond the scope of this letter. We would just add that the NMF tracker performance came close to the Cramér–Rao bound for tracking [2] in several cases, when we performed such an investigation; these studies will be published separately. Finally, in practical applications, detecting a track is only a part of the overall tracking procedure. The detection procedure described in this section and illustrated in Section III is performed many times, as new data are acquired. Detected short-term tracks (tracklets) are connected into long-term tracks. In this process, spurious events are discarded, and tracks not detected initially are detected at a later time. This procedure, however, is not a subject of this letter. Graphical representation of the described NMF architecture is shown in Fig. 1.

$$a_k = (\ldots)_h \sum_{n \in N} f(h|n)(\ldots)_n.$$  \hspace{1cm} (5)

Using these notations, parameters are computed at each iteration as

$$a_{k_0} = (\ldots)_h$$  \hspace{1cm} (6)

$$y_{0_0}(1)_h + v_{y_0}(t_0)_h = (y_{0_0})_h$$  \hspace{1cm} (7)

$$x_{0_0}(1)_h + v_{x_0}(t_0)_h = (x_{0_0})_h$$  \hspace{1cm} (8)

$$x_{0_0}(1)_h + v_{x_0}((D_0)_h + c(1)) = (x_{0_0})_h + c(D_0)_h.$$  \hspace{1cm} (9)
is that this number is linear in $N$ and in $H$, rather than combinatorial $\sim H^N$ like in MHT.

### III. Tracking Example

An application example of the NMF technique described previously is illustrated in Fig. 2, where detection and tracking are performed on targets below the clutter level. Fig. 2(a) shows true track positions in a 0.5 km $\times$ 0.5 km data set, while Fig. 2(b) shows the actual data available for detection and tracking. In this data, the target returns are buried in the clutter, with signal-to-clutter ratio of about $-2$ dB for amplitude and $-3$ dB for Doppler. Here, the data is displayed such that all six revisit scans are shown superimposed in the 0.5 km $\times$ 0.5 km area, 500 predetected signals per scan, and the brightness of each data sample is proportional to its measured Doppler value. Fig. 2(c)–(h) illustrates the evolution of the NMF model as it adapts during increasing iterations. Here, Fig. 2(c) shows the initial vague-fuzzy model, while Fig. 2(h) shows the model upon convergence at 20 iterations. Between Fig. 2(c) and (h), the NMF NN automatically decides how many model components are needed to fit the data, and simultaneously adapts the model parameters, including target track parameters. There are two types of models: one uniform model describing clutter (it is not shown), and linear track models with large uncertainty; the number of track models is determined from data as described in Section II. In Fig. 2(c) and (d), the NMF NN fits the data with one model, and uncertainty is somewhat reduced. Between Fig. 2(d) and (e), NMF uses more than one track model and decides that it needs two models to “understand” the content of the data. Fitting with two tracks continues until Fig. 2(f); between Fig. 2(f) and (g), a third track is added. Iterations stop at Fig. 2(h), when similarity stops increasing. Detected tracks closely correspond to the truth (Fig. 2(a)).

Fig. 2 illustrations are directly related to values of the model parameters described in Section II. The brightness of the figure is proportional to neural weights (3). Each track begins at $(x_0, y_0, t_0)$ and extends to $(x_0 + vx_0 t_N, y_0 + vy_0 t_N)$. Thickness of tracks is determined by standard deviation $\sigma$.

The number of neurons is dominated by the number of data neurons; their total number is $500 \times 6 = 3000$. The number of modeling neurons varied during iterations with the number of models, most of the time; as we can see in Fig. 2 frames, there were four modeling neurons, one for clutter and three for track models.

In this example, target signals are below clutter. A single scan does not contain enough information for detection. Detection should be performed concurrently with tracking, using several radar scans, and six scans are used. In this case, a standard multiple hypothesis tracking, evaluating all tracking association hypothesis, would require about $10^{1500}$ operations, a number too large for computation. Therefore, tracking requires strong signals, with about 15-dB signal-to-clutter ratio [1]. NMF successfully detected and tracked all three targets and required only $10^6$ operations, achieving about 18-dB improvement in signal-to-clutter sensitivity.

In case of low clutter (signal-to-clutter ratio above 15 dB), MHT can successfully detect tracks and in this case accuracy of MHT tracks is expected to be comparable to NMF, when tracks are well separated. If probability densities of radar measurements are known, such estimates will be unbiased. When tracks overlap, MHT is expected to produce biased results. The reason is that NMF estimation is not true maximum likelihood, but conditional on hypotheses about correct data-to-track assignments. For close tracks, assignments are biased, therefore, estimations are biased. If probability densities are used for similarities, MHT produces true maximum–likelihood estimation, including maximum–likelihood assignments, and therefore, they are expected to be unbiased (for many shapes of probability densities). However, this bias is a relatively small problem. The real advantage of NMF is its ability to track in heavy clutter, improving upon other tracking algorithms by orders of magnitude; in other words, solving previously unsolvable problems.

### IV. Future Directions

Future research will include feature-added tracking when—in addition to amplitude, position, and velocity—other characteristics of received signals are also used for improved associations between signals and track models. The NMF NN can naturally incorporate this additional information. Since association neural weights in NMF are functions of object models (1) any object feature can be included into the models and will be used for signal-model associations.
Other sources of information can be included. For example, coordinates of roads can be easily incorporated into the NMF procedure. For this purpose, road positions should be characterized by a probability density, depending on the known coordinates and expected errors. Then, similarities (2) can be modified by multiplying them by the probability densities of roads.

REFERENCES


I. INTRODUCTION

Time delays in neural signal processing are often unavoidable in the hardware implementation or sometimes desirable in engineering applications of neural networks (NNs). As time delays may cause undesirable instability and the stability of recurrent NNs is usually a prerequisite for successful applications, stability analysis of recurrent NNs with time delays is deemed necessary. In recent years, many new results on the stability analysis of various recurrent NNs with time delays have been reported; e.g., see [1]–[6].

In addition to time delays, the parameters of recurrent NNs are also subject to random and abrupt changes; i.e., the connection weights and biases of recurrent NNs change from one set to another randomly. Besides, in some applications such as real-time control and pattern recognition [7], more than one NN works cooperatively by switching from one to another. In these cases, NNs have to be modeled as switching systems. The stability of recurrent NNs with switching parameters also deserves in-depth analysis. For example, the stability of a switching Cohen–Grossberg NN [8] with mixed time-varying delays was recently analyzed in [6].

Noise-induced stabilization is an effective approach by which a dynamic system can be stabilized by using a proper white noise [9], [10]. Generally, it is thought that adding noise to an originally stable NN would cause the NN to be unstable. However, it will be shown in this letter that recurrent NNs with mixed time-varying delays and Markovian-switching parameters can be stabilized by using a suitable white noise, no matter whether it originally is stable or not, provided that the time delays are sufficiently small.

In this letter, we present new results on the stabilization of a general class of NNs by noise. A delay-dependent algebraic criterion on noise will be shown to stabilize the Cohen–Grossberg NN model [8] with mixed time-varying delays and Markovian-switching parameters (e.g., connection weights and biases are subject to changes according to a Markovian chain).

II. PRELIMINARIES

Throughout this letter, unless otherwise specified, let $\Omega \subseteq \mathcal{F} \subseteq \mathcal{F}_{\leq 0}, P$ be a complete probability space with a filtration $\{\mathcal{F}_{\tau}\}_{\tau \geq 0}$ satisfying the usual conditions (i.e., it is right continuous and contains all $\mathcal{F}$-null sets). Let $\omega(t)$ be a scalar Brownian motion defined on the probability space. Let $C([-\tau, 0], R^n)$ denote the family of continuous functions $\varphi$ from $[-\tau, 0]$ to $R^n$ with the norm $||\varphi|| = \sup_{t \geq 0 \leq \tau} |\varphi(t)|$, where $\tau > 0$. $|\cdot|$ is the Euclidean norm in $R^n$. Let $R_k = [0, \infty)$ and $\tau_k : R_k \to [0, \tau_k] (k = 2, 3)$ be a continuous function which stands for the time delay. The operator norm of matrix $A$ is denoted by $||A|| = \sup \{|Ax| : |x| = 1\}$ (without any confusion with $||x||$). Denote $C_{\mathcal{F}_{\tau}}([\tau, 0], R^n)$ as the family of all bounded, $\mathcal{F}_{\tau}$-measurable, $C([\tau, 0], R^n)$-valued random variables. Let $x(t)$ be a continuous $R^n$-valued stochastic process on $t \in [\tau, \infty)$, and let $x_{\tau} = \{x(t + \theta) : -\tau \leq \theta \leq 0\}$ for $t \geq 0$ which is regarded as a $C([-\tau, 0], R^n)$-valued stochastic process.