Neural Networks for Improved Tracking

Leonid I. Perlovsky, *Senior Member, IEEE*, and Ross W. Deming, *Member, IEEE*

***Abstract*—We have developed a neural network based upon modeling fields for improved object tracking. Models for 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 Radar, Neural Networks, Multi-target Tracking.**

I. Introduction

GROUND Moving Target Indication (GMTI) radar signals have been used for tracking since the 1990's [1]. When clutter is strong, so that signals are below clutter, detection and tracking is difficult. The Cramer-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 then 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 thirty 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 100^{100} 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, radar 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 radar 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 combining neural architecture and a model-based structure suitable for tracking a single target. In this article a neural architecture is developed for tracking multiple targets in clutter, while avoiding combinatorial complexity. The tracker is tailored in this article 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 theory NMF [8], each input neuron, $n = 1,\dots N,$ encodes four values: range and cross-range positions of cell n (x_n, y_n), and radar return parameters amplitude and Doppler (a_n, D_n); we denote $\mathbf{X}(n) = (x_n, y_n, a_n, D_n)$. Every radar return is also characterized by time t_n. A set of $\mathbf{X}(n)$ we call an input neural field, it is a set of bottom-up input signals. Top-down, or priming signals to these

Manuscript received September 11, 2006. The authors are thankful to AFOSR for supporting part of this research under the Lab. Task 05SN02COR, PM Dr. Jon Sjogren.

L. I. Perlovsky is a visiting scholar at Harvard University, and technical advisor at the Air Force Research Laboratory, Sensors Directorate. Harvard contact info: 33 Oxford St, Rm 336, Cambridge MA 02138, Tel. 617-496-1339; 617-495-7871; leonid@seas.harvard.edu. Air force contact info: AFRL/SNHE 80 Scott Drive, Hanscom AFB, MA 01731, Tel. 781-377-1728; Fax 781-377-8984; Leonid.Perlovsky@hanscom.af.mil.

R. W. Deming is a consultant for General Dynamics IT, Inc., working at the Air Force Research Laboratory, Sensors Directorate, SNHE branch, 80 Scott Drive, Hanscom AFB, MA 01731, ross.deming@hanscom.af.mil.

neurons are generated by models, a clutter-model and track-models, $\boldsymbol{M}_h(S_h,n)$, enumerated by index $h = 1,\dots H$. Each model is characterized by its parameters, S_h. In this paper we consider targets moving with constant velocities, (vx_h, vy_h):

$$\boldsymbol{M}_h(S_h,n) = (x0_h + vx_h\, t_n,\ y0_h + vy_h\, t_n,\ a_h,\ D_h) \qquad (1)$$

These models predict expected positions of targets. The neural network compares these predictions to the data and generates parameter update signals as described later. Model parameters $S_h = (x0_h,\ vx_h,\ y0_h,\ vy_h,\ a_h,\ D_h)$ characterize position, velocity, average amplitude, and Doppler of the target, also $D_h = vx_h$. These parameters are not known and are to be estimated through the internal dynamics of the NMF neural network.

Interaction between bottom-up and top-down signals is determined by neural weights. First, define similarity measures between bottom-up signals $\boldsymbol{X}(n)$ and top-down signals $\boldsymbol{M}_h$,

$$\ell(\mathrm{n|h}) = (2\pi)^{-2}(\det \mathrm{C_h})^{-0.5} \cdot \exp\{-0.5\ (\mathbf{X}(\mathrm{n}) - \mathbf{M}_\mathrm{h}^{\mathrm{T}})\mathrm{C_h}^{-1}\ (\mathbf{X}(\mathrm{n}) - \mathbf{M}_\mathrm{h})\ \}. \qquad (2)$$

Here we use a Gaussian function with the mean $\boldsymbol{M}_h$ and a diagonal covariance $C_h = diag(\sigma x_h^2, \sigma y_h^2, \sigma a_h^2, \sigma D_h^2)$. Second, compute neural weights,

$$f(h|n) = r(h)\ \ell(n|h) \Big/ \sum_{h' \in H} r(h')\ \ell(n|h'). \qquad (3)$$

Here rates $r(h)$ are the relative numbers of radar returns associated with model h. For object models we assume that a single radar return is received from each scan; this is compatible with a low setting of threshold for radar signal detections. The number of radar scans, k, is known, therefore, $r(h) = k$ are known quantities. The clutter model we denote by $h = 1$, it is characterized by average amplitude, Doppler, and rate,

$$M_1(S_1, n) = (a_1, D_1), \quad r(1) = (N - H\,k)/N, \qquad (4)$$

Model predictions are computed by modeling neurons, therefore the total number of neurons is the number of data points N plus the number of models H.

NMF neural network iterations [8] start with an initial guess using known minimum and maximum values of coordinates, *Xmin, Xmax, Ymin, Ymax.* The maximum absolute values of velocities in either direction (x, y) are 100 mi/h, or 44.7 m/s, *Vmin* = -44.7 m/s, *Vmax* = 44.7 m/s. Amplitude is normalized to $Amin = 0$, $Amax = 1$. Target initial parameter values are taken randomly between the minimum and maximum values. Clutter initial values are taken as zero for Doppler and *0.5* for amplitude. Initial values of the standard deviations, $\sigma_{initial}$, are estimated using all radar returns, using same values for all models. Next, NMF computes neural weights (3), then computes new values of the model parameters. Model parameter equations are written using the following notation for a standard neural operation of computing a weighted sum,

$$<\dots>_h = \sum_{n \in N} f(h|n)\ (\dots)_n. \qquad (5)$$

Using these notations, parameters are computed at each iteration as

$$a_h = \ < a_n >_h. \qquad (6)$$

$$\begin{aligned} y0_h < 1 >_h + vy_h < t_n >_h &= \ < y_n >_h, \\ y0_h < t_n >_h + vy_h < t_n^2 >_h &= \ < y_n\, t_n >_h. \end{aligned} \qquad (7)$$

$$\begin{aligned} x0_h < 1 >_h + vx_h < t_n >_h &= \ < x_n >_h, \\ x0_h < t_n >_h + vx_h\, (\ < t_n^2 >_h + c < 1 >\ ) &= \\ &= \ < x_n\, t_n >_h + c < D_n >_h. \end{aligned} \qquad (8)$$

Here, $c = \sigma x_h^2 ./\sigma D_h^2$. For the unknown parameters, $y0_h$ and vy_h, (7) is a linear system of equations; similarly (8) is a two-dimensional linear system of equations for $x0_h$ and vx_h. After parameters are computed as above, standard deviations are computed, as follows:

$$\sigma = \ \sigma_{initial}\ exp\{-7\ i\ /\ IT\ \} + \sigma_{final}. \qquad (9)$$

Here, $\sigma_{initial}$ and σ_{final} are initial and final values of standard deviation. The initial value is defined using standard statistical procedure [11] from all available data. The final value is defined by the radar measurement errors. The exponential factor is defined so that at the end of iterations the first item is small; i is the iteration number, and IT is the total number of iterations; it is set from experience. Usually, it is between *10* and *100*; results are not very sensitive and few trials are sufficient. Number of tracks is also an unknown parameter estimated from the data. Here we describe a simple procedure that we used in the example in the following section. We start with a larger number of track models than is actually expected. After every five iterations, if estimated tracks come closer to each other than two standard deviations, one of them is removed and reinitialized randomly. After convergence ($i = IT$), for each track we compute detection measures defined as a local log-likelihood ratio computed using returns within two standard deviations $\{n'\}$ of each track:

$$LLR(h) = \sum_{n' \in N'} [\ \log \ell(n'|h) - \log \ell(n'|1)\ ]. \qquad (10)$$

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 the next section took *20* iterations. Since similarity is a highly non-linear function, regular convergence to the global maximum can not be expected. The local rather than global convergence sometimes presents an irresolvable

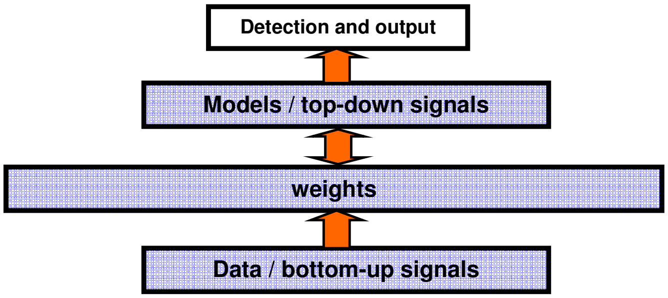


Fig. 1. Graphical representation of the NMF architecture.

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 smoothes local maxima. Second, according to the above description (before equation (10)), a large number of tracks is used initially; many of them are re-initiated several times. Therefore if a particular real track is not "captured" after few iterations, it will be captured at a later iteration, after track re-initialization. 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 log-likelihood ratio (10) will be low and spurious events are discarded. A detailed characterization of performance usually requires operating curves [12], plots of probability of detection vs. 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 Cramer-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 the next section 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 paper. Graphical representation of the described NMF architecture is shown in Fig. 1.

The computational complexity of the procedure described in this section is proportional to the number of data points and the number of tracks, $const*N*H$. The *const* here accounts for the number of iterations, and for complexity of procedures described by (1) through (9). Typical numbers are discussed in the next section. The principal theoretical moment 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 above 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 0.5km * 0.5km 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 –2dB for amplitude and –3dB for Doppler. Here, the data is displayed such that all six revisit scans are shown superimposed in the 0.5km * 0.5km area, *500* pre-detected signals per scan, and the brightness of each data sample is proportional to its measured Doppler value. Figs. 2(c)-2(h) illustrate 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 (c) and (h) the NMF neural network 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 the previous section. In (c) and (d), the NMF neural network fits the data with one model, and uncertainty is somewhat reduced. Between (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 *2* tracks continues until (f); between (f) and (g) a third track is added. Iterations stop at (h), when similarity stops increasing. Detected tracks closely correspond to the truth (a).

Fig.2 illustrations are directly related to values of the model parameters described in previous section. The brightness of the figure is proportional to neural weights (3). Each track begins at $(x0_h, y0_h)$ and extends to $(x0_h + vx_h\ t_N, y0_h + vy_h\ t_N)$. Thickness of tracks is determined by standard deviation σ.

The number of neurons is dominated by the number of data neurons; their total number is $500x6 = 3000$. 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 *4* modeling neurons, *1* for clutter and *3* 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 a 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, NMF is expected to produce biased results. The reason is that NMF estimation is not true

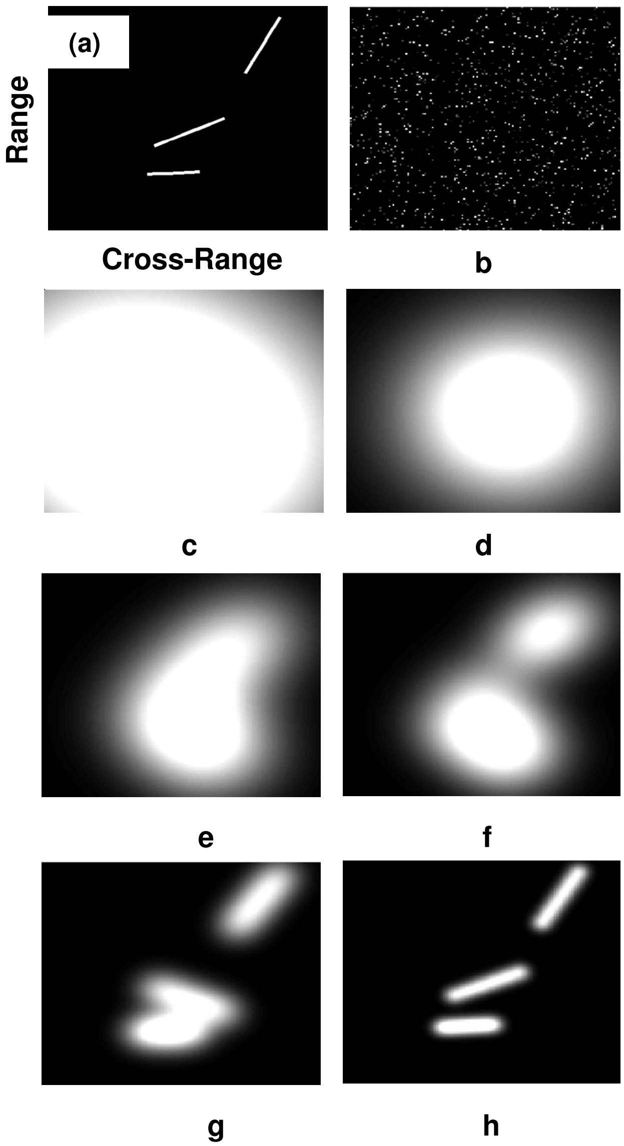


Fig. 2. Detection and tracking objects below clutter using NMF: (a) true track positions in 0.5km * 0.5km data set; (b) actual data available for detection and tracking. Evolution of the NMF neural network is illustrated in (c) – (h), where (c) shows the initial, uncertain model and (h) shows the model upon convergence after 20 iterations. Note the close agreement between the converged model (h) and the truth (a).

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, NMF produces true maximum likelihood estimation, including maximum likelihood assignments, and therefore are expected to be unbiased (for many shapes of probability densities). But, let me emphasize again, 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 neural network 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

[1] J. Jones, J. Bradstreet, M. Kozak, T. Hughes, and M. Blount, "Ground moving target tracking and exploitation performance measures," Pentagon Report A269234, approved for public release, 2004.

[2] L. I. Perlovsky, "Cramer-Rao bound for tracking in clutter and tracking multiple objects," *Pattern Recognition Letters*, vol. 18, pp. 283-288, 1997.

[3] R. E. Bellman, *Adaptive Control Processes*, Princeton, NJ: Princeton Univ. Press, 1961.

[4] L. I. Perlovsky, "Conundrum of computational complexity," *IEEE Trans. PAMI*, vol. 20, pp. 666-670, 1998.

[5] N. Wiener, *Cybernetics*, New York, NY: Wiley, 1948.

[6] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Trans. ASME, Journal of Basic Engineering*, vol. D82, pp. 35-45, 1960.

[7] R. A. Singer, R. G. Sea, and R. B. Housewright, "Derivation and evaluation of improved tracking filters for use in dense multitarget environments," *IEEE Trans. Information Theory*, vol. IT-20, pp. 423-432, 1974.

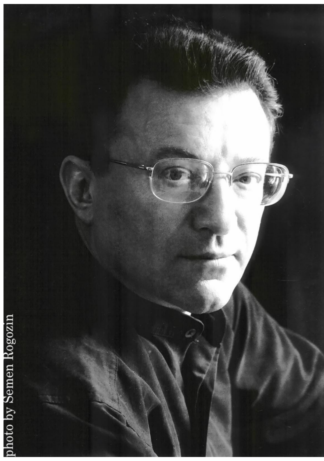
[8] L. I. Perlovsky, *Neural Networks and Intellect*, New York, NY: Oxford Univ. Press, 2001.

[9] Y. Bar-Shalom and E. Tse, "Tracking in a cluttered environment with probabilistic data association," *Automatica*, vol. 11, pp. 451-460, 1975.

[10] B. Widrow, "Adaptive Sample-Data Systems – a Statistical Theory of Adaptation," 1959 WESCON Convention Record, Part 4, pp. 74-85, 1959.

[11] Anderson, Sweeney, and Williams, *Statistics for Business and Economics*, 9th ed., Academic Internet Publishers, 2006.

[12] See (2007, June) [Online] Available: http://en.wikipedia.org/wiki/Receiver_operating_characteristic for references and discussion.



Leonid Perlovsky (M'86-SM'81) Ph.D.'74 in theoretical and mathematical physics, Institute for Mathematics, Novosibirsk and Joint Institute for Nuclear Research, Dubna, Moscow, Russia.

He is Visiting Scholar at Harvard University, Cambridge, MA and Principal Research Physicist and Technical Advisor at the Air Force Research Laboratory, Hanscom AFB, MA, Program Manager for DOD Semantic Web program and other research projects. From 1985 to 1999, he served as Chief Scientist at Nichols Research. Previously Professor at Novosibirsk University and New York University, principal in commercial startups developing tools for text understanding, biotechnology, and financial predictions. His company predicted the market crash following 9/11 a week before the event, detecting activities of Al Qaeda traders, and later helped SEC tracing the leads. He delivered invited keynote plenary talks and tutorial lectures worldwide, published more then 270 papers, 7 book chapters, and authored a monograph "Neural Networks and Intellect," Oxford University Press, 2001 (currently in the 3rd printing). Two more books are scheduled to appear in 2007.

Dr. Perlovsky is a member of INNS, Chairs IEEE Boston Computational Intelligence Chapter, organizes conferences on Computational Intelligence, serves as Associate Editor for IEEE Transactions on Neural Networks, Editor-at-Large for "Natural Computations" and Editor-in-Chief for "Physics of Life Reviews." He received the IEEE Distinguished Member of Boston Section Award 2005; the US AF Research Lab Charles Ryan Memorial Award for Basic Research 2007; and International Neural Network Society Gabor Award for achievement in engineering/applications 2007.

Ross Deming (M'05) received the B.S. degree from Cornell University in 1985, the M.S. degree from the University of Vermont in 1993, and the Ph.D. degree from Northeastern University in 1996, all in electrical engineering.

He was previously employed at General Electric, Inc. and Teradyne, Inc., as an electrical engineer, and at Nichols Research Corp. and Witten Technologies, Inc., as a senior scientist. Currently, he is a consultant for General Dynamics, Inc., where he is working on radar and optical signal processing problems for the Air Force Research Laboratory.

While at Witten Technologies (WTI), he was part of a small team that won, among other awards, the Wall Street Journal's 2004 Technology Innovation Award (1st place software category, honorable mention overall) for its technology that creates detailed images of objects and conditions underground using radar. He is also coauthor on a patent for this radar system.