Neural Networks 23 (2010) 435–451

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

Biased ART: A neural architecture that shifts attention toward previously disregarded features following an incorrect prediction

Gail A. Carpenter *, Sai Chaitanya Gaddam

Department of Cognitive and Neural Systems, Boston University, 677 Beacon Street, Boston, MA 02215, USA

ARTICLE INFO

Article history:
Received 2 April 2009
Accepted 17 July 2009

Keywords:
Adaptive resonance theory
ART
ARTMAP
Featural biasing
Supervised learning
Top-down/bottom-up interactions

ABSTRACT

Memories in Adaptive Resonance Theory (ART) networks are based on matched patterns that focus attention on those portions of bottom-up inputs that match active top-down expectations. While this learning strategy has proved successful for both brain models and applications, computational examples show that attention to early critical features may later distort memory representations during online fast learning. For supervised learning, biased ARTMAP (bARTMAP) solves the problem of over-emphasis on early critical features by directing attention away from previously attended features after the system makes a predictive error. Small-scale, hand-computed analog and binary examples illustrate key model dynamics. Two-dimensional simulation examples demonstrate the evolution of bARTMAP memories as they are learned online. Benchmark simulations show that featural biasing also improves performance on large-scale examples. One example, which predicts movie genres and is based, in part, on the Netflix Prize database, was developed for this project. Both first principles and consistent performance improvements on all simulation studies suggest that featural biasing should be incorporated by default in all ARTMAP systems. Benchmark datasets and bARTMAP code are available from the CNS Technology Lab Website: http://techlab.bu.edu/bART/.

1. Introduction

During learning, Adaptive Resonance Theory (ART) models encode attended featural subsets called critical feature patterns. With winner-take-all coding, when a novel exemplar activates an established category only the features of the bottom-up input that are also in the top-down critical feature pattern remain active in working memory. The network hereby focuses attention on a subset of the input, ignoring other incoming features as not relevant to the currently active category. If the top-down/bottom-up pattern meets a matching criterion, the learned critical feature pattern sharpens, shedding features not represented in the current input.

The strategy of learning attended critical feature patterns, rather than basing memories on whole bottom-up inputs, has proved successful both in models of cognitive information processing and in applications of unsupervised ART and supervised ARTMAP systems. However, focusing on features that were critical early in learning may lead a system later to pay too much attention to these features. Computational examples show that, for certain input sequences, such undue featural attention can distort system memories and reduce test accuracy. If training inputs are repeatedly presented, an ARTMAP system will correct these errors — but real-time learning may not afford such repeat opportunities before action is required.

Biased ARTMAP (bARTMAP) solves the problem of over-emphasis on early critical features by directing attention away from previously attended features after the system makes a predictive error. A variety of examples demonstrate that bARTMAP performance is consistently better than that of fuzzy ARTMAP. Small-scale, hand-computed analog and binary examples illustrate key model dynamics. Two-dimensional simulation examples demonstrate the evolution of bARTMAP memories as they are learned online. Benchmark simulations show that featural biasing also improves performance on large-scale examples. The Boston remote sensing image example (Carpenter, Martens, & Ogas, 2005) has been used in previous studies. A second example, which predicts movie genres and is based, in part, on the Netflix Prize database, was developed for this project. Both benchmark datasets and biased ARTMAP software are available from the CNS Technology Lab Website (http://techlab.bu.edu/bART/).

For a given training input, biased ARTMAP tracks attended features that have led to predictive errors, and reduces activation of these features during search. Bias strength is controlled by a free parameter λ, with the network reducing to the unbiased system (fuzzy ARTMAP) when $\lambda = 0$. For a given application, an optimal value of λ can be determined by validation, but setting λ equal to a default value of 10 produces near-optimal results on small-scale and large-scale computational examples. Improvements in

* Corresponding author. Tel.: +1 617 353 9483; fax: +1 617 353 7755.
E-mail address: gail@bu.edu (G.A. Carpenter).

0893-6080/$ – see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.neunet.2009.07.025

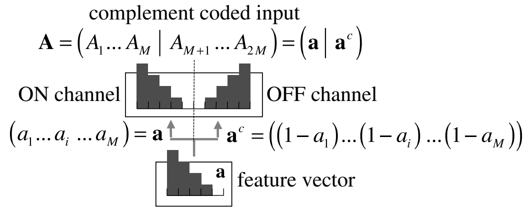

Fig. 1. Complement coding transforms an M-dimensional feature vector $\mathbf{a}$ into a $2M$-dimensional system input vector $\mathbf{A}$. A complement-coded input represents both the degree to which a feature i is present (a_i) and the degree to which that feature is absent ($1 - a_i$).

test accuracy are accompanied by reduced overlap of the category boxes that geometrically represent network memories, with little or no increase in network size. All examples use the same default ARTMAP parameters, with winner-take-all coding, fast learning, and maximum generalization. In a fast-learning system, long-term memory variables reach their asymptotes on each input trial.

2. ART and ARTMAP

ART neural networks model real-time prediction, search, learning, and recognition. ART networks serve both as models of human cognitive information processing (Carpenter, 1997; Grossberg, 1999, 2003) and as neural systems for technology transfer (Caudell, Smith, Escobedo, & Anderson, 1994; Lisboa, 2001; Parsons & Carpenter, 2003).

Design principles derived from scientific analyses and design constraints imposed by targeted applications have jointly guided the development of many variants of the basic networks, including fuzzy ARTMAP (Carpenter, Grossberg, Markuzon, Reynolds, & Rosen, 1992), ARTMAP-IC (Carpenter & Markuzon, 1998), and Gaussian ARTMAP (Williamson, 1998). One distinguishing characteristic of different ARTMAP models is the nature of their internal code representations. Early ARTMAP systems, including fuzzy ARTMAP, employ winner-take-all coding, whereby each input activates a single category node during both training and testing. When a node is first activated during training, it is permanently mapped to its designated output class.

Starting with ART-EMAP (Carpenter & Ross, 1995), ARTMAP systems have used distributed coding during testing, which typically improves predictive accuracy while avoiding the design challenges inherent in the use of distributed code representations during training. In order to address these challenges, distributed ARTMAP (Carpenter, 1997; Carpenter, Milenova, & Noeske, 1998) introduced a new network configuration, new learning laws, and even a new unit of long-term memory, replacing traditional weights with adaptive thresholds (Carpenter, 1994).

Comparative analysis of the performance of ARTMAP systems on a variety of benchmark problems has led to the identification of a *default ARTMAP* network (Carpenter, 2003), which features simplicity of design and robust performance in many application domains. Default ARTMAP employs winner-take-all coding during training and distributed coding during testing within a distributed ARTMAP network configuration. With winner-take-all coding during testing, default ARTMAP reduces to the version of fuzzy ARTMAP that is used here as the basis of comparison with biased ARTMAP. However, the biased ARTMAP mechanism is a small modular addition to the ART orienting subsystem, and could be readily added to any other version of the network.

2.1. Complement coding: Learning both absent and present features

ART and ARTMAP employ a preprocessing step called *complement coding* (Fig. 1), which models the nervous system's ubiquitous computational design known as *opponent processing* (Hurvich & Jameson, 1957). Balancing an entity against its opponent, as in agonist–antagonist muscle pairs, allows a system to act upon relative quantities, even as absolute magnitudes may vary unpredictably. In ART systems, complement coding (Carpenter, Grossberg, & Rosen, 1991) is analogous to retinal ON-cells and OFF-cells (Schiller, 1982). When the learning system is presented with a set of feature values $\mathbf{a} \equiv (a_1 \ldots a_i \ldots a_M)$, complement coding doubles the number of input components, presenting to the network both the original feature vector $\mathbf{a}$ and its complement $\mathbf{a}^c$.

Complement coding allows an ART system to encode within its critical feature patterns of memory features that are consistently *absent* on an equal basis with features that are consistently *present*. Features that are sometimes absent and sometimes present when a given category is learning becomes uninformative with respect to that category. Since its introduction, complement coding has been a standard element of ART and ARTMAP networks, where it plays multiple computational roles, including input normalization. However, this device is not particular to ART, and could, in principle, be used to preprocess the inputs to any type of system.

To implement complement coding, component activities a_i of a feature vector $\mathbf{a}$ are scaled so that $0 \le a_i \le 1$. For each feature i, the ON activity a_i determines the complementary OFF activity $(1 - a_i)$. Both a_i and $(1 - a_i)$ are represented in the $2M$-dimensional system input vector $\mathbf{A} = (\mathbf{a} \mid \mathbf{a}^c)$ (Fig. 1). Subsequent network computations operate in this $2M$-dimensional input space. In particular, learned weight vectors $\mathbf{w}_J$ are $2M$-dimensional.

2.2. ARTMAP search and match tracking

The ART matching process triggers either learning or a parallel memory search (Fig. 2). When search ends, the learned memory may either remain the same or incorporate new information from matched portions of the current input. While this dynamic applies to arbitrarily distributed activation patterns at the coding field F_2, the code will here be described as a single active category node J in a winner-take-all system.

Before ARTMAP makes an output class prediction, the bottom-up input $\mathbf{A}$ is matched against the top-down learned expectation, or critical feature pattern, that is read out by the active node (Fig. 2b). The matching criterion is set by a parameter ρ called *vigilance*. Low vigilance permits the learning of abstract prototype-like patterns, while high vigilance requires the learning of specific exemplar-like patterns. When a new input arrives, vigilance equals a baseline level $\bar{\rho}$. Baseline vigilance is set equal to zero by default in order to maximize generalization. Vigilance rises after the system has made a predictive error. The internal control process that determines how far ρ must rise in order to correct the error is called *match tracking* (Carpenter, Grossberg, & Reynolds, 1991). As vigilance rises, the network is required to pay more attention to how well top-down expectations match the current bottom-up input.

Match tracking (Fig. 3) forces an ARTMAP system not only to reset its mistakes but to learn from them. With match tracking, fast learning, and winner-take-all coding, each ARTMAP network passes the Next Input Test, which requires that, if a training input were re-presented immediately after a learning trial, it would directly activate the correct output class, with no predictive errors or search. Match tracking simultaneously implements the design goals of maximizing generalization and minimizing predictive error without requiring the choice of a fixed matching criterion. ARTMAP memories thereby include both broad and specific pattern classes, with the latter typically formed as