ARTICLE IN PRESS
JID:PLA AID:22596 /SCO Doctopic: Biological physics [m5G; v 1.134; Prn:15/05/2014; 11:31] P.1 (1-5)
Physics Letters A ●●● (●●●●) ●●●–●●●

Contents lists available at ScienceDirect

Physics Letters A

www.elsevier.com/locate/pla

Designing neural networks that process mean values of random variables

M.J. Barber [a], J.W. Clark [b,c]

[a] AIT Austrian Institute of Technology, Innovation Systems Department, 1220 Vienna, Austria
[b] Department of Physics and McDonnell Center for the Space Sciences, Washington University, St. Louis, MO 63130, United States
[c] Centro de Ciências Matemáticas, Universidade de Madeira, 9000-390 Funchal, Portugal

ARTICLE INFO

Article history:
Received 3 January 2014
Received in revised form 19 April 2014
Accepted 23 April 2014
Available online xxxx
Communicated by C.R. Doering

Keywords:
Neural networks
Probabilistic models
Bayesian networks
Bayesian inference
Neural information processing
Population coding

ABSTRACT

We develop a class of neural networks derived from probabilistic models posed in the form of Bayesian networks. Making biologically and technically plausible assumptions about the nature of the probabilistic models to be represented in the networks, we derive neural networks exhibiting standard dynamics that require no training to determine the synaptic weights, that perform accurate calculation of the mean values of the relevant random variables, that can pool multiple sources of evidence, and that deal appropriately with ambivalent, inconsistent, or contradictory evidence.

1. Introduction

Artificial neural networks are noted for their ability to learn functional relationships from observed data. Unfortunately, a trained neural network is typically a black box, so that it can be quite difficult to determine what function is actually represented by the network. In numerous cases, neural networks have been related to probabilistic models, with either the trained network retrospectively given a probabilistic interpretation or the training process itself explicitly based on a probabilistic strategy. Alternatively, a constructive approach can be taken to exploring representation of probabilistic models in neural networks, encoding pre-specified probabilistic models into network weights. The key challenge of such an approach is to produce reasonable neural networks, allowing a suitably broad class of probabilistic models to be encoded into neural networks with recognizable architectures and dynamics. Towards this end, we formulate and characterize an encoding method that handles a restricted class of probabilistic models and allows calculation, without training, of neural networks that accurately process the mean values of the relevant random variables with the usual neural activation of a weighted sum of the neural inputs transformed with a nonlinear activation function.

Constructive or "top-down" approaches to artificial and natural neural information processing receive impetus from current conceptual thrusts in theoretical neurobiology. Specifically, it has been proposed [1] that cortical circuits (as neural populations) perform Bayesian statistical inference, encoding and processing information about pertinent analog variables in terms of their probability densities (PDs). This hypothesis supports a theoretical framework for understanding diverse results of neurobiological experiments, and a practical framework for the design of recurrent neural network models that implement a broad variety of information-processing functions [2–4].

Probabilistic formulations of neural information processing have been explored along a number of avenues since the strong resurgence of research in artificial neural networks in the 1980s and the parallel genesis of computational neuroscience. Early developments focused on the prospects of "stochastic machines" [5], notably Boltzmann machines [6], sigmoid belief networks [7], and Helmholtz machines [8]. These networks are composed of stochastic processing units that occupy one of two possible states in a probabilistic manner. Learning rules for stochastic machines enable such systems to model the underlying probability distribution of a given data set.

A seminal analysis by Anderson and Abrahams [9] in 1987 presaged the introduction of Bayesian probability theory into the formal description of neural information processing, by demonstrating that the original Hopfield neural network implements, in effect, Bayesian inference on analog quantities in terms of PDs (see also [10]).

http://dx.doi.org/10.1016/j.physleta.2014.04.065
0375-9601/© 2014 Published by Elsevier B.V.

As in the present work, which stems from that of Refs. [2,11,3,4], Zemel, Dayan, and Pouget [12] have investigated population coding of probability distributions, but with different representations and dynamics than those we consider here. Several extensions [13] of the Bayesian probabilistic framework envisioned by these authors have been developed that feature information propagation between interacting neural populations. (See also [14]; for pertinent reviews see [15] and especially [16].)

The connection between neural networks and probabilistic models represented specifically as Bayesian networks [17,18] has been explored along two main directions. In one approach, the neural network architecture and activation dynamics are specified, and a learning rule is applied that endeavors to capture the appropriate Bayesian network in the synaptic weights based on observed patterns [7,19,20].

In the second approach, a prescribed Bayesian network is transformed into a neural network using an encoding process formulated in terms of probability densities of analog variables [2,11]. While specific Bayesian networks are readily captured in this approach, the neural architecture and neuronal dynamics that arise from the encoding need not match those of traditional models. In particular, instead of the usual weighted sum of neural activation values passed through a nonlinear activation function, the encoding process can produce neural networks that involve multiplicative interactions between neural activities. Even so, the modular nature of cortical processing [21–23] is well suited to such a strategy, in which cortical areas are taken to collectively represent the joint probability density over several variables. These neural "problem-solving modules" can be mapped in a relatively direct fashion onto the nodes of a Bayesian belief network, giving rise to a class of neural network models that have been termed *neural belief networks* [2,11].

In contrast, studies of Eliasmith and Anderson based on population-temporal coding [3,4] provide strong evidence that the modeling of low-level sensory processing and output motor control do not require such a sophisticated representation: for these functions, manipulation of mean values instead of full PDs is generally sufficient. Further, the representations can be simplified to deal with vector spaces describing the mean values instead of function spaces describing the PDs.

In the present work, we develop neural networks processing mean values of analog variables as a specialized form of the more general neural belief networks. We begin with a brief summary of the relevant properties of Bayesian networks in Section 2. We describe a procedure for generating and evaluating the neural networks in Section 3, and apply this procedure to salient examples in Section 4.

2. Bayesian networks

Bayesian networks, or Bayesian belief nets [17,18], are directed acyclic graphs that represent probabilistic models. Fig. 1 provides an illustrative example. Each node represents a random variable, while the arcs signify the presence of dependence between the linked variables. The strengths of these influences are defined using conditional probabilities. We additionally take the direction of a given link to indicate the direction of causality (or, more simply, relevance), with an arc pointing from cause to effect; in this form, the Bayesian network is also called a causal network.

Multiple sources of evidence about the random variables are conveniently handled using Bayesian networks. The belief, or degree of confidence, in particular values of the random variables is determined as the likelihood of the value given evidentially support provided to the network. There are two types of support that arise from the evidence: predictive support, which propagates from cause to effect along the direction of the arc, and retrospective support, which propagates from effect to cause, opposite to the direction of the arc.

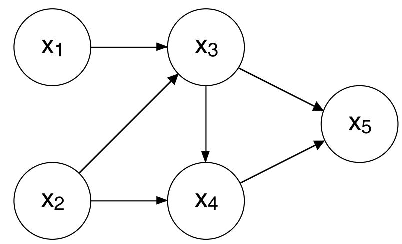

Fig. 1. A Bayesian network. Evidence about any of the random variables influences the likelihood of the remaining random variables. In a straightforward terminology, the node at the tail of an arrow is a parent of the child node at the head of the arrow, e.g., X_4 is a parent of X_5 and a child of both X_2 and X_3. From the structure of the graph, we can see the conditional independence relations in the probabilistic model. For example, X_5 is independent of X_1 and X_2, given X_3 and X_4.

Bayesian networks have two properties that we will find very useful, both of which stem from the dependence relations shown by the graph structure. First, the value of a node X is not dependent upon all of the other graph nodes. Rather, it depends only on a subset of the nodes, called a Markov blanket of X, which separates node X from all the other nodes in the graph. The Markov blanket of interest to us is readily determined from the graph structure. It is comprised of the union of the direct parents of X, the direct successors of X, and all direct parents of the direct successors of X. The second property is that the joint probability over the random variables x_μ is decomposable as

$$P(x_1, x_2, \ldots, x_n) = \prod_{\mu=1}^{n} P\big(x_\mu \,\big|\, \mathrm{Pa}(x_\mu)\big), \tag{1}$$

where $\mathrm{Pa}(x_\mu)$ denotes the (possibly empty) set of direct-parent nodes of X_μ. This decomposition stems from repeated application of Bayes' rule and from the structure of the graph.

3. Neural network model

We will develop neural networks from the set of marginal distributions $\{\rho(x_\mu; t)\}$ so as to best match a desired probabilistic model $\rho(x_1, x_2, \ldots, x_D)$ over the set of random variables x_μ, which are organized as a Bayesian network. One or more of these variables must be specified as evidence in the Bayesian network. To facilitate the development of general update rules, we do not distinguish between evidence and non-evidence nodes in our notation.

Our general approach will be to minimize the difference between a probabilistic model $\rho(x_1, x_2, \ldots, x_D)$ and an estimate of the probabilistic model $\hat{\rho}(x_1, x_2, \ldots, x_D)$. For the estimate, we utilize

$$\hat{\rho}(x_1, x_2, \ldots, x_D) = \prod_{\alpha=1}^{D} \rho(x_\alpha; t). \tag{2}$$

This is a so-called naive estimate, wherein the random variables are assumed to be independent. In principle, one may systematically improve upon this estimate by including successively higher-order correlations in an exact product decomposition of the density $\rho(x_1, x_2, \ldots, x_D)$, as in Ref. [10], at the expense of introducing multiplicative interactions between neurons.

Further constraints are placed on the probabilistic model and representation so as to produce neural networks with the desired dynamics. The first assumption we make is that the populations of neurons only need to accurately encode the mean values of the random variables, rather than the complete densities. We take the