Neurocomputing 188 (2016) 3–11

Contents lists available at ScienceDirect

Neurocomputing

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

Modeling time series data with deep Fourier neural networks ☆

Michael S. Gashler *, Stephen C. Ashmore

University of Arkansas, Fayetteville, AR 72701, USA

ARTICLE INFO

Article history:
Received 1 July 2014
Received in revised form
11 November 2014
Accepted 23 January 2015
Available online 2 December 2015

Keywords:
Neural networks
Time-series
Curve fitting
Extrapolation
Fourier decomposition

ABSTRACT

We present a method for training a deep neural network containing sinusoidal activation functions to fit to time-series data. Weights are initialized using a fast Fourier transform, then trained with regularization to improve generalization. A simple dynamic parameter tuning method is employed to adjust both the learning rate and the regularization term, such that both stability and efficient training are achieved. We show how deeper layers can be utilized to model the observed sequence using a sparser set of sinusoid units, and how non-uniform regularization can improve generalization by promoting the shifting of weight toward simpler units. The method is demonstrated with time-series problems to show that it leads to effective extrapolation of nonlinear trends.

1. Introduction

Finding an effective method for predicting nonlinear trends in time-series data is a long-standing unsolved problem with numerous potential applications, including weather prediction, market analysis, and control of dynamical systems. Fourier decompositions provide a mechanism to make neural networks with sinusoidal activation functions fit to a training sequence [25,38,44], but it is one thing to fit a curve to a training sequence, and quite another to make it extrapolate effectively to predict future nonlinear trends. We present a new method utilizing deep neural network training techniques to transform a Fourier neural network into one that can facilitate practical and effective extrapolation of future nonlinear trends.

Much of the work in machine learning thus far has focused on modeling static systems. These are systems with behavior that depends only on a set of inputs, and do not change behavior over time. One example of a static system is medical diagnosis. An electronic diagnostic tool could be designed to evaluate the symptoms that a patient reports and attempt to label the patient with a corresponding medical condition. Presumably, a correct diagnosis depends only on the reported symptoms. Certainly, time plays a factor in the prevalence of medical conditions, but the change is often either too slow, or too unpredictable to warrant consideration in the model. Thus, time is effectively an irrelevant feature in this domain.

When time is a relevant feature, such as with climate patterns, market prices, population dynamics, and control systems, one possible modeling approach is to simply treat time as yet another input feature. For example, one could select a curve and adjust its coefficients until the curve approximately fits with all the observations that have been made so far. Predictions could then be made by feeding in an arbitrary time to compute a future prediction.

Nonlinear curve-fitting approaches tend to be very effective at interpolation, predicting values among those for which it was trained, but they often struggle with extrapolation, predicting values outside those for which it was trained. Because extrapolation requires predicting in a region that is separated from all available samples, any superfluous complexity in the model tends to render predictions very poor. Thus far, only very simple models, such as linear regression, have been generally effective at extrapolating trends in time-series data. Finding an effective general method for nonlinear extrapolation remains an open challenge.

We use a deep artificial neural network to fit time-series data. Artificial neural networks are not typically considered to be simple models. Indeed, a neural network with only one hidden layer has been shown to be a universal function approximator [9]. Further, deep neural networks, which have multiple hidden layers, are used for their ability to fit to very complex functions. For example, they have been found to be very effective in the domain of visual recognition [6,23,22,12,37]. It would be intuitive to assume, therefore, that deep neural networks would be a poor choice of

☆This work was supported in part by the National Science Foundation under Grants ARI #0963249, MRI #0959124 (Razor), EPS #0918970 (CI TRAIN), and a grant from the Arkansas Science and Technology Authority, managed by the Arkansas High Performance Computing Center.

* Corresponding author.
E-mail addresses: mgashler@uark.edu (M.S. Gashler), scashmor@uark.edu (S.C. Ashmore).
URL: http://csce.uark.edu/~mgashler (M.S. Gashler).

http://dx.doi.org/10.1016/j.neucom.2015.01.108
0925-2312/

model for extrapolation. However, our work demonstrates that a careful approach to regularization can enable complex models to extrapolate effectively, even with nonlinear trends.

This document is laid out as follows: Section 2 summarizes related work. Section 3 gives a description of our approach for extrapolation with time-series data. Section 4 reports results that validate our approach. Finally, Section 5 summarizes the contributions of this work.

2. Related work

Many papers have surveyed the various techniques for using neural networks and related approaches to model and forecast time-series data [11,21,42,14,41,10]. Therefore, in this section, we will review only the work necessary to give a high-level overview of how our method fits among the existing techniques, and defer to these other papers to complete an exhaustive survey of related techniques. At a high level, the various approaches for training neural networks to model time-series data may be broadly categorized into three major groups. Fig. 1 illustrates the basic models that correspond with each of these groups.

The most common, and perhaps simplest, methods for time-series prediction involve feeding sample values from the past into the model to predict sample values in the future [14,1]. (See Fig. 1A.) These methods require no recurrent connections, and can be implemented without the need for any training techniques specifically designed for temporal data. Consequently, these can be easily implemented using many available machine learning toolkits, not just those specifically designed for forecasting time-series data. These convenient properties make these methods appealing for a broad range of applications. Unfortunately, they also have some significant limitations: The window size for inputs and predictions must be determined prior to training. Also, they essentially use recent observations to represent state, and they are particularly vulnerable to noise in the observed values.

A more sophisticated group of methods involves neural networks with recurrent connections [29]. These produce their own internal representation of state (see Fig. 1B). This enables them to learn how much to adjust their representations of state based on observed values, and hence operate in a manner more robust against noisy observations.

Existing methods for training the weights of recurrent neural networks can be broadly divided into two categories: Those based on nonlinear global optimization techniques, such as evolutionary optimization [13,34,3], and those based on descending a local error gradient, such as Backpropagation Through Time [27,39] or Real-Time Recurrent Learning [31,26]. Unfortunately, in practice, evolutionary optimization tends to be extremely slow, and it is unable to yield good results with many difficult problems [35,34]. Gradient-based methods tend to converge faster than global optimization methods, but they are more susceptible to problems with local optima. With recurrent neural networks, local optima are a more significant problem than with regular feed-forward networks [8]. The recurrent feedback can create chaotic responses in their error surfaces, and can distribute local optima in poor regions of the space.

In early instantiations, recurrent neural networks struggled to retain internal state over long periods of time because the logistic activation functions typically used with neural networks tend to squash the activations with each time step. Long Short Term Memory architectures address this problem by using only linear units with the recurrent loops [19]. This advance has made recurrent neural networks much more capable of modeling time-series data. Recent advances in deep neural network learning have also helped to improve the training of recurrent neural networks [30,7,16,18].

The third, and the most relevant, group of methods for forecasting time-series data is regression followed by extrapolation. Extrapolation with linear regression has long been a standard method for forecasting trends. Extrapolating nonlinear trends, however, has generally been ineffective. The general model, shown in Fig. 1C, is very simple, but training it in a manner that will make generalizing predictions can be extremely challenging. For this reason, this branch of time-series forecasting has been much less studied, and remains a relatively immature field. Our work attempts to jump-start research in this branch by presenting a practical method for extrapolating nonlinear trends in time-series data.

The idea of using a neural network that can combine basis functions to reconstruct a signal, and initializing its weights with a Fourier transform, has been previously proposed [33,25], and more recently methods for training them have begun to emerge [38,44]. These studies, however, do not address the important practical issues of stability during training and regularizing the model to promote better generalization. By contrast, our work treats the matter of representing a neural network that can combine basis functions as a solved problem, and focuses on the more challenging problem of refining these networks to achieve reliable nonlinear extrapolation.

Many other approaches, besides Fourier neural networks, have been proposed for fitting to time-series data. Some popular approaches include wavelet networks [43,4,2,17,36,5], and support vector machines [20,24,32].

3. Algorithm description

Our approach uses a deep artificial neural network with a mixture of activation functions. We train the network using stochastic gradient descent [40]. Each unit in the artificial neural network uses one of three activation functions, illustrated in Fig. 2, sinusoid: $f(x) = \sin(x)$, softplus: $f(x) = \log_e(1+e^x)$, or identity: $f(x) = x$. Using the "identity" activation function creates a linear unit, which is only capable of modeling linear components in the data. Nonlinear components in the data require a nonlinear activation function. The softplus units enable the network to fit to non-repeating nonlinearities in the training sequence. The sinusoid units enable the network to fit to repeating nonlinearities in the data.

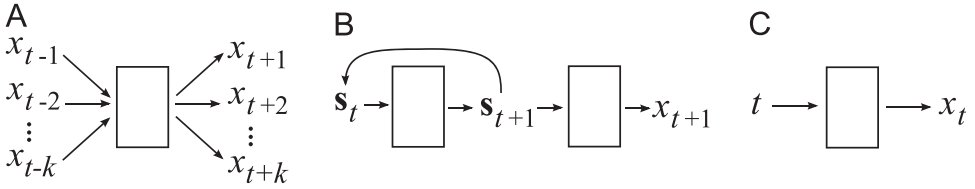

Fig. 1. Methods that use neural networks for time-series prediction can be broadly grouped into three general approaches. Approach **A** uses samples from the past to predict samples in the future. Approach **B** uses recurrent connections to remember and modify an internal representation of state. Approach **C** fits a curve to the time-series data and uses extrapolation to make predictions. There are many variations on each of these approaches. Our work makes advances within group **C**.