Abstract

We propose Significance-Offset Convolutional Neural Network, a deep convolutional network architecture for multivariate time series regression. The model is inspired by standard autoregressive (AR) models and gating mechanisms used in recurrent neural networks. It involves an AR-like weighting system, where the final predictor is obtained as a weighted sum of sub-predictors, while the weights are data-dependent functions learnt through a convolutional network. The architecture was designed for applications on asynchronous time series and hence is evaluated on such datasets: a hedge fund proprietary dataset of over 2 million quotes for a credit derivative index, an artificially generated noisy autoregressive series and household electricity consumption dataset. The proposed architecture achieves promising results as compared to convolutional and recurrent neural networks.

1. Introduction

In this paper we examine the capabilities of convolutional neural networks (CNNs) (Lecun et al., 1998) in modeling the conditional mean of the distribution of future observations; in other words, the problem of autoregression. We focus on time series with multivariate and noisy signal. In particular, we work with financial data which has received limited public attention from the deep learning community and for which nonparametric methods are not commonly applied. Financial time series are particularly challenging to predict due to their low signal-to-noise ratio (cf. applications of Random Matrix Theory in econophysics (Laloux et al., 2000; Bun et al., 2017)) and heavy-tailed distributions (Cont, 2001). Moreover, the predictability of financial market returns remains an open problem and is discussed in many publications (cf. efficient market hypothesis (Fama, 1970)).

It is a common case that with financial data the same information (e.g. value of an asset) is observed from different sources (e.g. financial news, analysts, portfolio managers in hedge funds, market-makers in investment banks) in irregular moments of time. Each of these sources may have a different bias and noise with respect to the original signal that needs to be recovered. Moreover, these sources are usually strongly correlated and lead-lag relationships are possible (e.g. a market-maker with more clients can update its view more frequently and precisely than one with fewer clients). Therefore, the significance of each of the available past observations might be dependent on some other factors that can change in time. Hence, the traditional econometric models such as AR, VAR, VARMA (Hamilton, 1994) might not be sufficient. Yet their relatively good performance motivates coupling such linear models with deep neural networks that are capable of learning highly nonlinear relationships.

For these reasons, we propose Significance-Offset Convolutional Neural Network, a Convolutional Network extension of standard autoregressive models (Sims, 1972; 1980) equipped with nonlinear weighting mechanism. We also provide empirical evidence on its competitiveness to popular convolutional and recurrent architectures.

2. Related work

2.1. Time series forecasting

Reading through recent proceedings of the main machine learning venues (e.g. ICML, NIPS, AISTATS, UAI), one can notice that time series are often forecast using Gaussian processes (Petelin et al., 2011, Tobar et al., 2015, Hwang et al., 2016), especially when time series are irregularly sampled (Cunningham et al., 2012, Li & Marlin, 2016).

On the other hand, deep neural networks have recently surpassed results from most of the existing literature in many fields (Schmidhuber, 2015): computer vision (Krizhevsky et al., 2012), audio signal processing and speech recognition (Sak et al., 2014), natural language processing (NLP) (Bengio et al., 2003, Collobert & Weston, 2008, Grave et al., 2016, Jozefowicz et al., 2016). Although sequence modeling in NLP, i.e. prediction of the next character or word, is related to our forecasting problem, the nature of the sequences
3. Motivation

Time series observed in irregular moments of time cause significant difficulties for learning algorithms. Gaussian processes provide useful theoretical framework capable of handling asynchronous data; however, due to assumed Gaussianity they are inappropriate for financial datasets, which often follow fat-tailed distributions ((Cont (2001)). On the other hand, even prediction of simple autoregressive time series may involve highly nonlinear functions when sampled irregularly.

We often deal with multivariate time series whose dimensions are observed separately and asynchronously. This adds even more difficulty to assigning appropriate weights to the past values, even if the underlying data structure is linear. Furthermore, appropriate representation of such series might be not obvious. As an alternative to aligning observations at some chosen frequency[1] we might consider representing separate dimensions as a single one with dimension and duration indicators as additional features. Figure 3 presents this approach, which is going to be at the core of the proposed SOCNN architecture.

![representation of asynchronous series](image)

Figure 1. Data representation for the asynchronous series. Consecutive observations are stored together as a single value series, regardless of which series they belong to; this information, however, is stored in indicator features, alongside durations between observations.

For these reasons we shall consider a model that combines simple autoregressive approach with neural network in order to allow learning meaningful data-dependent weights

\[
\mathbb{E}[x_n | \{ x_{n-m}, m = 1, \ldots, M \}] = \sum_{m=1}^{M} \alpha_1 (x_{n-m}) \cdot x_{n-m},
\]

where \((\alpha_m)_{m=1}^M\) satisfying \(\alpha_1 + \cdots + \alpha_M \leq 1\) are modeled using neural network. To allow more flexibility and cover

---

1Which is highly inefficient in case when durations have varying magnitudes.
4. Model Architecture

Suppose that we are given a multivariate time series \((x_n)_n \subset \mathbb{R}^d\) and we aim to predict the conditional future values of a subset of elements of \(x_n\)

\[
y_n = \mathbb{E}[x_n^I | \{x_{n-m}, m = 1, 2, \ldots\}], \quad (3)
\]

where \(I = \{i_1, i_2, \ldots, i_d\} \subset \{1, 2, \ldots, d\}\) is a subset of features of \(x_n\). Let \(x_n^{-M} = (x_{n-m})_{m=1}^M\). We consider the following estimator of \(y_n\)

\[
\hat{y}_n^{(i)} = \sum_{m=1}^M \left[ F(x_n^{-M}) \otimes \sigma(S(x_n^{-M})) \right]_{im}, i \in 1, 2, \ldots, d_I, \quad (4)
\]

where

- \(F, S : \mathbb{R}^{d \times M} \to \mathbb{R}^{d_I \times M}\) are neural networks described below,
- \(\sigma\) is a normalized activation function independent on each row, i.e.
  \[
  \sigma((a_1^T, \ldots, a_d^T)^T) = (\sigma(a_1)^T, \ldots, \sigma(a_d)^T)^T \quad (5)
  \]
  for any \(a_1, \ldots, a_d \in \mathbb{R}^M\) and \(\sigma\) such that \(\sigma(a)^T 1_M = 1\) for any \(a \in \mathbb{R}^M\).
- \(\otimes\) is Hadamard (element-wise) matrix multiplication.

The summation in (4) goes over the columns of the matrix in bracket; hence the \(i\)-th element of the output vector \(\hat{y}_n\) is a linear combination of the \(i\)-th row of the matrix \(F(x_n^{-M})\). We are going to consider \(S\) to be a fully convolutional network (composed solely of convolutional layers) and \(F\) of the form

\[
F(x_n^{-M}) = W \otimes \left[ \text{off}(x_{n-m}) + x_n^I \right]_{m=1}^M \quad (6)
\]

where \(W \in \mathbb{R}^{d_I \times M}\) and \(\text{off} : \mathbb{R}^d \to \mathbb{R}^{d_I}\) is a multilayer perceptron. In that case \(F\) can be seen as a sum of projection \((x \mapsto x^I)\) and a convolutional network with all kernels of length 1. Equation (4) can be rewritten as

\[
\hat{y}_n = \sum_{m=1}^M \left[ W_m \otimes \left( \text{off}(x_{n-m}) + x_n^I \right) \right] \otimes \left[ S_m(x_n^{-M}) \right], \quad (7)
\]

where \(W_m, S_m(\cdot)\) are \(m\)-th columns of matrices \(W\) and \(S(\cdot)\).

We will call the proposed network a Significance-Offset Convolutional Neural Network (SOCNN), while \text{off} and \text{significance} (sub)networks. The network scheme is shown in Figure 2. Note that when \text{off} \equiv 0 and \(\sigma \equiv 1\) the model simplifies to the collection of \(d_I\) separate \(AR(M)\) models for each dimension.

**Interpretation of the components**

Note that the form of Equation (7) enforces the separation of temporal dependence (obtained in weights \(W_m\)), the local significance of observations \(S_m (S\) as a convolutional network is determined by its filters which capture local dependencies and are independent on the relative position in time) and the predictors \(\text{off}(x_{n-m})\) that are completely independent on position in time. This provides some amount of interpretability of the fitted functions and weights. For instance, each of the past observations provides a single estimate of the target variable through the offset network.
5. Experiments

We evaluate the proposed model on a financial dataset of bid/ask quotes sent by several market participants active in the credit derivatives market and an artificially generated synchronous and asynchronous dataset and household electricity consumption dataset available from UCI repository \cite{Lichman2013}. In each case, the objective is to predict one step ahead conditional on 60 past observations. For quotes dataset, we formed 6 separate tasks, each of which involved prediction of the next quote by one of the 6 most active market participants.

Performance is compared with VAR model, CNN, single- and multi-layer LSTM \cite{Hochreiter1997} and 25-layer ResNet \cite{He2015}. The benchmark networks were designed so that they handle exactly the same input data, have comparable numbers of parameters and similar structure to the proposed model; hyperparameters were chosen in a grid search \cite{Maas2013}. To analyze importance of the components of SOCNN, we consider offset subnetwork with 1 and 10 layers. Mean squared error was used as a performance measure and training objective in all cases.

\footnote{The dataset contains 2.1 million quotes from 28 different market participants. Each quote is characterized by 31 features: the offered price, 28 indicators of the quoting source, the direction indicator (the quote refers to either a buy or a sell offer) and duration from the previous quote.}

\footnote{We consider 4 artificial series of length 10,000 and dimensionality of 16 and 64. The synchronous series consist of \( K \in \{16, 64\} \) noisy copies (‘sources’) of the same univariate autoregressive base series, observed together at random times; the noise of each copy is of different type. The asynchronous series are sampled from the respective synchronous ones by randomly choosing one of their dimensions at each time step; therefore each step consists of a value at sampled dimension, the indicator of sampled dimension and duration since last observation.}

\footnote{Electricity dataset contains measurements of 7 different quantities related to electricity consumption in a single household, recorded every minute for 47 months, yielding over 2 million observations. Since we aim to focus on asynchronous time-series, we alter it so that a single observation contains only a value of one of the seven features, while durations between consecutive observations range from 1 to 7 minutes. The regression aim is to predict all of the features at the next time step. The original dataset is available at UCI Machine Learning Learning Repository website \url{https://archive.ics.uci.edu/}.}

\footnote{The code for experiments and simulated series are available online at \url{https://github.com/mhinkowski/ntimeseries}.}

\footnote{Architecture details: for SOCNN, CNN, ResNet and LSTM we used respectively 10, 10, 25 and from 1 up to 3 layers. Number of channels/memory cells per layer was equal to 16 or 32 (half of these for SOCNN due to its two-leg structure) and was selected through grid search, together with dropout rate (0 or .5) and gradient clipping (0 or .001). In convolutional networks we used 3 max pooling layers (except fully-convolutional SOCNN) while the kernel sizes alternated between 1 and 3. LeakyReLU activation \( \sigma(x) = \max(x, ax) \) \cite{Maas2013} with leak rate of \( a = .1 \) was used in all layers except the top ones.}

Results

Table I presents the results from artificial and electricity datasets. The proposed networks outperform significantly the benchmark networks on the asynchronous, electricity and quotes datasets. For the synchronous datasets, on the other hand, SOCNN almost matches the results of the benchmark. Such similar performance could have been anticipated - the correct weights of the past values in synchronous artificial datasets are presumably less nonlinear than in asynchronous case. For this reason, the significance network’s potential is not fully utilized.

Table I. Detailed results. For each model, we present the mean squared error obtained on the out-of-sample test set. The best results for each dataset are marked by bold font. For quotes dataset the presented values are averaged mean-squared errors from 6 separate prediction tasks, normalized according to the error obtained by VAR model.

<table>
<thead>
<tr>
<th>model</th>
<th>VAR</th>
<th>CNN</th>
<th>ResNet</th>
<th>LSTM</th>
<th>SOCNN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Synchronous 16</td>
<td>0.841</td>
<td>0.152</td>
<td>0.150</td>
<td>0.152</td>
<td>0.154</td>
</tr>
<tr>
<td>Synchronous 64</td>
<td>0.364</td>
<td>0.028</td>
<td>0.028</td>
<td>0.028</td>
<td>0.029</td>
</tr>
<tr>
<td>Asynchronous 16</td>
<td>0.577</td>
<td>0.040</td>
<td>0.032</td>
<td>0.027</td>
<td>0.017</td>
</tr>
<tr>
<td>Asynchronous 64</td>
<td>0.318</td>
<td>0.041</td>
<td>0.046</td>
<td>0.050</td>
<td>0.032</td>
</tr>
<tr>
<td>Electricity</td>
<td>0.729</td>
<td>0.366</td>
<td>0.359</td>
<td>0.463</td>
<td>0.158</td>
</tr>
<tr>
<td>Quotes</td>
<td>1.000</td>
<td>0.975</td>
<td>3.565</td>
<td>3.696</td>
<td>0.420</td>
</tr>
</tbody>
</table>

We also observe that the depth of the offset network has negligible or negative impact on the results achieved by the SOCNN network. This means that the significance network is crucial for the SOCNN’s performance and obtaining proper weights for the past observations is much more challenging than getting good predictors from the single past values of the series.

For quotes dataset, the proposed model was the best one for all the tasks and the only one to always beat the VAR model. Surprisingly, for each of the other networks it was difficult to excel the benchmark set by simple linear model.

We also found benchmark networks to have unstable test loss during training in some cases, despite convergence of the training error. Especially, for one of the tasks LSTM and ResNet obtained very high test errors.

Model robustness

We analyze robustness of the model by checking its susceptibility to additional noise in the input. Considering 16-dimensional asynchronous dataset, for each datapoint \( (x_{n-M}, y_n) \) we add noise of magnitude to every 5th of the past observations \( \epsilon = x_{n-5k} + \xi \) and observe how the prediction errors change for each trained model, for varying \( \xi \). Figure 2 presents results of this experiment for SOCNN, CNN and LSTMs.

6. Conclusion and discussion

In this article, we proposed a weighting mechanism which, coupled with convolutional networks, forms a new neural
Figure 3. Changes in the prediction mean squared error with respect to the varying noise in 20% of input steps. LSTM1 and LSTM2 denote respectively one- and two-layer LSTMs. SOCNN appears to be more robust and adaptive to unseen data (note the uncentered curves for the other models for the test set), and less prone to overfitting, as opposed to CNN. The dotted lines represent the respective average offset and significance outputs for noisy inputs. Results are averaged over 3000 random train/test samples. The proposed model can be further extended by adding intermediate weighting layers of the same type in the network architecture for time series prediction that proved successful in tested asynchronous regression tasks.

Finally, we aim at testing the performance of the proposed architecture on other real-life datasets with relevant characteristics. We observe that there exists a strong need for common ‘econometric’ datasets benchmark and, more generally, for time series (stochastic processes) regression.

7. Acknowledgements

Authors would like to thank Hellebore Capital Ltd. for providing data for the experiments. M.B. thanks Engineering and Physical Sciences Research Council (EPSRC) for partial funding of this research.

References


Autoregressive Convolutional Neural Networks for Asynchronous Time Series


