Supplementary material to "Autoregressive Convolutional Neural Networks for Asynchronous Time Series" - Appendices

Mikołaj Bińkowski [1 2] **Gautier Marti** [2 3] **Philippe Donnat** [2]

A. Nonlinearity in the asynchronously sampled autoregressive time series

Lemma 1. *Let $X(t)$ be an AR(2) time series given by*

$$X(t) = aX(t-1) + bX(t-2) + \varepsilon(t), \tag{1}$$

where $(\varepsilon(t))_{t=1,2,\dots}$ are i.i.d. error terms. Then

$$\mathbb{E}[X(t)|X(t-1), X(t-k)] = a_k X(t-1) + b_k X(t-k), \tag{2}$$

for any $t > k \geq 2$, where a_k, b_k are rational functions of a and b.

Proof. The proof follows a simple induction. It is sufficient to show that

$$w_k X(t) = v_k X(t-1) + b^{k-1} X(t-k) + E_k(t), \quad k \geq 2, \tag{3}$$

where $w_k = w_k(a,b), v_k = v_k(a,b)$ are polynomials given by

$$(w_2, v_2) = (1, a) \tag{4}$$

$$(w_{k+1}, v_{k+1}) = (-v_k, -(bw_k + av_k)), \quad k \geq 2, \tag{5}$$

and $E_k(t)$ is a linear combination of $\{\varepsilon(t-i), i = 0, 1, \dots, k-2\}$. Basis of the induction is trivially satisfied via 1. In the induction step, we assume that 3 holds for k. For $t > k+1$ we have $w_k X(t-1) = v_k X(t-2) + b^{k-1} X(t-k-1) + E_k(t-1)$. Multiplying sides of this equation by b and adding $av_k X(t-1)$ we obtain

$$\begin{aligned}(av_k + bw_k)X(t-1) = v_k(aX(t-1) + bX(t-2)) \\ + b^k X(t-k-1) + bE(t-1).\end{aligned} \tag{6}$$

Since $aX(t-1) + bX(t-2) = X(t) - \varepsilon(t)$ we get

$$\begin{aligned}-v_{k+1}X(t-1) = -w_{k+1}X(t) + b^k X(t-k-1) \\ + [bE_k(t-1) - v_k\varepsilon(t)]\end{aligned} \tag{7}$$

As $E_{k+1}(t) = bE_k(t-1) - v_k\varepsilon(t)$ is a linear combination of $\{\varepsilon(t-i), i = 0, 1, \dots, k-1\}$, the above equation proves 3 for $k = k+1$.

□

B. Robustness of the proposed architecture

To see how robust each of the networks is, we add noise terms to part of the input series and evaluate them on such datapoints, assuming unchanged output. We consider varying magnitude of the noise terms, which are added only to the selected 20% of past steps at the value dimension[1]. Formally the procedure is following:

1. Select randomly $N_{obs} = 6000$ observations (X_n, y_n) (half of which coming from training set and half from test set).
2. Add noise terms to the observations $\widetilde{X_n}^p := X_n + \Xi_n \cdot \gamma_p$, for $\{\gamma_p\}_{p=1}^{128}$ evenly distributed on $[-6\sigma, 6\sigma]$, where σ is a standard deviation of the differences of the series being predicted and
$$(\Xi_n)_{tj} = \begin{cases} \xi_n \sim \mathcal{U}[0,1] & \text{if } j = 0, t \in [0, 5, \dots, 55] \\ 0 & \text{otherwise.} \end{cases} \tag{8}$$
3. For each p evaluate each of the trained models on dataset $\left\{\widetilde{X_n}^p, y_n\right\}_{n=1}^{N_{obs}}$, separately for n's originally coming from training and test sets.

C. Artificial data generation

We simulate a multivariate time series composed of K noisy observations of the same autoregressive signal. The simulated series are constructed as follows:

*Equal contribution [1]Department of Mathematics, Imperial College London, London, UK [2]Hellebore Capital Limited, London, UK [3]Laboratoire d'informatique, Ecole Polytechnique, Palaiseau, France. Correspondence to: Mikołaj Bińkowski <mikbinkowski at gmail dot com>.

Proceedings of the 35^{th} International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018.

[1]The asynchronous series has one dimension representing the *value* of the quote, one representing *duration* and others representing indicators of the *source*. See C for details.

1. We simulate univariate stationary AR(10) time series x with randomly chosen weights.
2. The series is copied K times and each copy $x^{(k)}$ is associated with a separate noise process $\varepsilon^{(k)}$. We consider Gaussian or Binomial noise of different scales; for each copy it is either added to or multiplied by the initial series ($x^{(k)} = x + \varepsilon^{(k)}$ or $x^{(k)} = x \times \varepsilon^{(k)}$).
3. We simulate a random time process T where differences between consecutive events are iid exponential random variables.

4a. The final series is composed of K noisy copies of the original process observed at times indicated by the random time process, and a duration between observations.

4b. At each time $T(t)$ indicated by the random time process T, one of the noisy copies k is drawn and its value at this time $x^{(k)}_{T(t)}$ is selected to form a new noisy series x^*. The final multivariate series is composed of x^*, the series of durations between observations and K indicators of which observation was drawn at each time.

Assume that $(x_t)_{t=1,2,\dots}$ is a stationary $AR(\nu)$ series and consider the following (random) noise functions

$$\begin{aligned}
\varepsilon_0(x, c, p) &= x + c(2\epsilon - 1), && \epsilon \sim \text{Bernoulli}(p),\\
\varepsilon_1(x, c, p) &= x(1 + c(2\epsilon - 1)), && \epsilon \sim \text{Bernoulli}(p),\\
\varepsilon_2(x, c, p) &= x + c\epsilon, && \epsilon \sim \mathcal{N}(0, 1),\\
\varepsilon_3(x, c, p) &= x(1 + c\epsilon), && \epsilon \sim \mathcal{N}(0, 1). \quad (9)
\end{aligned}$$

Note that argument p of ε_2 and ε_3 is redundant and was added just for notational convenience.

Let $N_t \sim \text{Exp}(\lambda)$ be a series of i.i.d. exponential random variables with some constant rate λ and let $T(t) = \sum_{s=1}^{t} \lceil N_s + 1 \rceil$. Then $T(t)$ is a strictly increasing series of times, at which we will observe the noisy observations.

Let $p_1, p_2, \dots, p_K \in (0, 1)$ and define

$$X^{(k)}_t := \begin{cases} \varepsilon_{k(\text{mod } 4)}(x_{T(t)}, 2^{-\lfloor k/8 \rfloor}, p_k), & k = 1, \dots, K,\\ T(t), & k = K + 1. \end{cases} \quad (10)$$

Let $I(t)$ be a series of i.i.d. random variables taking values in $\{1, 2, \dots, K\}$ such that $\mathbb{P}(I(t) = K) \propto q^K$ for some $q > 0$. Define

$$\bar{X}^{(k)}_t := \begin{cases} 1, & k \le K \text{ and } k = I(t),\\ 0, & k \le K \text{ and } k \ne I(t),\\ X^{(I(t))}_t, & k = K + 1,\\ T(t), & k = K + 2. \end{cases} \quad (11)$$

We call $\{X_t\}_{t=1}^N$ and $\{\bar{X}_t\}_{t=1}^N$ *synchronous* and *asynchronous* time series, respectively. We simulate both of the processes for $N = 10,000$ and each $K \in \{16, 64\}$.

D. Household electricity dataset

The original dataset has 7 features: global active power, global reactive power, voltage, global intensity, sub-metering 1, sub-metering 2 and sub-metering 3, as well as information on date and time. We created asynchronous version of this dataset in two steps:

1. Deterministic time step sampling. The durations between the consecutive observations are periodic and follow a scheme [1min, 2min, 3min, 7min, 2min, 2min, 4min, 1min, 2min, 1min]; the original observations in between are discarded. In other words, if the original observations are indexed according to time (in minutes) elapsed since the first observation, we keep the observations at indices n such that $n \mod 25 \equiv k \in [0, 1, 3, 6, 13, 15, 17, 21, 22, 24]$.
2. Random feature sampling. At each remaining time step, we choose one out of seven features that will be available at this step. The probabilities of the features were chosen to be proportional to $[1, 1.5, 1.5^2, 1.5^6]$ and randomly assigned to each feature before sampling (so that each feature has constant probability of its value being available at each time step.

At each time step the sub-sampled dataset is 10-dimensional vector that consists information about the time, date, 7 indicator features that imply which feature is available, and the value of this feature. The length of the sub-sampled dataset is above 800 thousand, i.e. 40% of the original dataset's length.

The schedule of the sampled timesteps and available features is attached in the *data* folder in the supplementary material.

E. Results

E.1. Detailed results for Quotes dataset

Table 1 presents the detailed results for the Quotes dataset.

E.2. Offset and significant weights in Electricity dataset

In Figure 1 we visualize significance and offset activations for three input series, from the network trained on *electricity* dataset. Each row represents activations corresponding to past values of a single feature.

Table 1. Detailed results for each prediction task for the quotes dataset. Each task involves prediction of the next quote by one of the banks. Numbers represent the mean squared errors on out-of-sample test set.

task	CNN	VAR	LSTM	Phased LSTM	ResNet	SOCNN
bank A	0.993	1.123	0.999	0.893	1.086	**0.530**
bank B	1.225	2.116	1.673	0.701	31.598	**0.613**
bank C	3.208	3.952	2.957	2.666	3.805	**0.617**
bank D	3.634	4.134	3.436	1.877	4.635	**0.649**
bank E	3.558	4.367	3.344	2.136	3.717	**1.154**
bank F	8.541	8.150	7.880	3.362	8.274	**1.553**
bank G	0.248	0.278	0.132	0.855	1.462	**0.063**
bank I	4.777	4.853	3.933	3.016	4.936	**0.400**
bank J	1.094	1.172	1.097	1.007	1.216	**0.773**
bank K	2.521	4.307	2.573	3.894	4.731	**0.926**
bank L	1.108	1.448	1.186	1.357	1.312	**0.743**
bank M	1.743	1.816	1.741	**0.941**	1.808	1.271
bank N	3.058	3.232	2.943	**1.123**	3.229	1.509
bank O	0.539	0.532	0.420	0.860	0.566	**0.218**
bank P	0.447	0.354	0.470	0.627	0.510	**0.283**

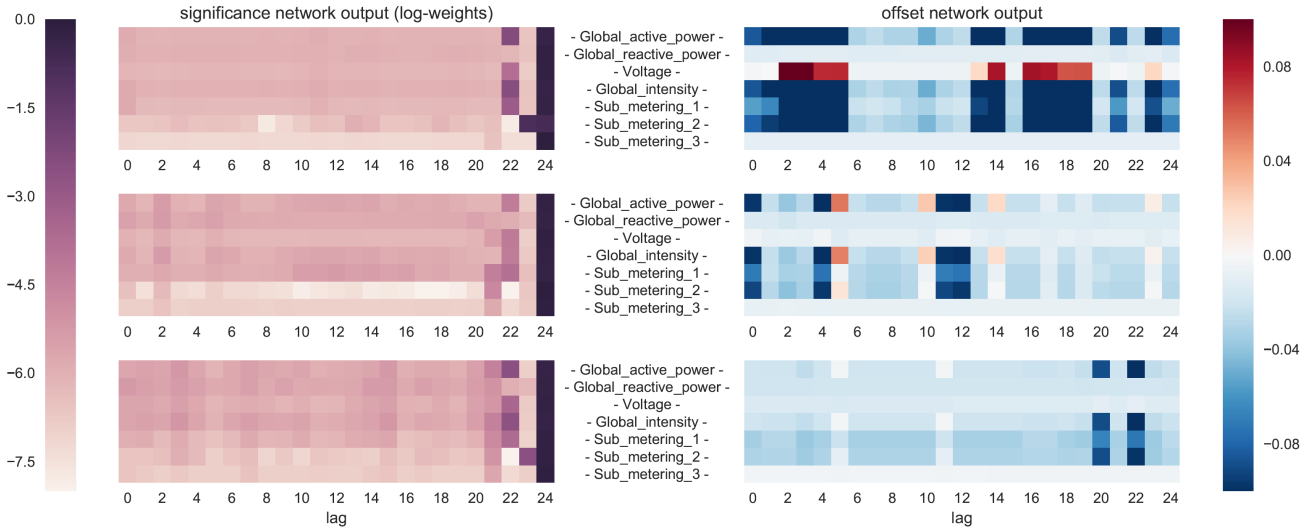

Figure 1. Activations of the *significance* and *offset* sub-networks for the network trained on Electricity dataset. We present 25 most recent out of 60 past values included in the input data, for 3 separate datapoints. Note the log scale on the left graph.