Autoregressive Convolutional Neural Networks for Asynchronous Time Series

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

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

[1]Imperial College London, UK [2]Hellebore Capital Ltd., London, UK [3]Ecole Polytechnique, Palaiseau, France. Correspondence to: Mikołaj Bińkowski <mikbinkowski@gmail.com>.

ICML 2017 Time Series Workshop, Sydney, Australia, 2017. Copyright 2017 by the author(s)

is too dissimilar to allow using the same cost functions and architectures. Same applies to the adversarial training proposed by (Mathieu et al., 2016) for video frame prediction, as such approach favors *most plausible* scenarios rather than outputs *close* to all possible outputs, while the latter is usually required in financial time series due to stochasticity of the considered processes.

Literature on deep learning for time series forecasting is still scarce (cf. (Gamboa, 2017) for a recent review). Literature on deep learning for *financial* time series forecasting is even scarcer though interest in using neural networks for financial predictions is not new (Mozer, 1993; McNelis, 2005). More recent papers include (Sirignano, 2016) who used 4-layer perceptrons in modeling price change distributions in Limit Order Books, and (Borovykh et al., 2017) who applied WaveNet architecture of (van den Oord et al., 2016a) to several short univariate and bivariate time-series (including financial ones). Despite the claim of applying deep learning, (Heaton et al., 2016) use autoencoders with a single hidden layer to compress multivariate financial data. Besides these and claims of secretive hedge funds, no promising results or innovative architectures were publicly published so far, to the best of authors' knowledge.

2.2. Gating and weighting mechanisms

Gating mechanisms for neural networks were first proposed by (Hochreiter & Schmidhuber, 1997) and proved essential in training recurrent architectures (Jozefowicz et al., 2016) due to their ability to overcome the problem of vanishing gradient. In general, they can be expressed as

$$f(x) = c(x) \otimes \sigma(x), \tag{1}$$

where f is the output function, c is a 'candidate output' (usually a nonlinear function of x), $\otimes$ is an element-wise matrix product and $\sigma : \mathbb{R} \to [0, 1]$ is a sigmoid nonlinearity that controls the amount of the output passed to the next layer (or to further operations within a layer). Appropriate compositions of functions of type 1 lead to the popular recurrent architectures such as LSTM (Hochreiter & Schmidhuber, 1997) and GRU (Chung et al., 2014).

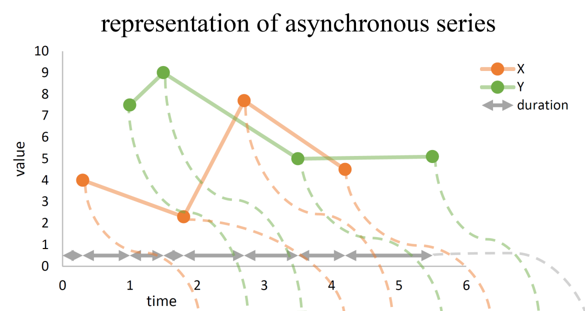
A similar idea was recently used in construction of highway networks (Srivastava et al., 2015) which enabled successful training of deeper architectures. (van den Oord et al., 2016b) and (Dauphin et al., 2016) proposed gating mechanisms (respectively with hyperbolic tangent and linear 'candidate outputs') for training deep convolutional neural networks.

The proposed gating system aims at weighting a number of different 'candidate predictors' and therefore is most closely related to the *softmax gating* used in MuFuRU (Multi-Function Recurrent Unit, (Weissenborn & Rocktäschel, 2016)) and attention networks (Cho et al., 2015).

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.



value	4.0	7.5	9.0	2.3	7.7	5.0	4.5	5.1
X indicator	1	0	0	1	1	0	1	0
Y indicator	0	1	1	0	0	1	0	1
duration	.3	.7	.5	.3	.9	.6	.7	1.3

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

$$\mathbb{E}[x_n | \{x_{n-m}, m = 1, \ldots, M\}] = \sum_{m=1}^{M} \alpha_1(x_{n-m}) \cdot x_{n-m}, \tag{2}$$

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

[1] Which is highly inefficient in case when durations have varying magnitudes.

situations when e.g. observed values of x are biased, we should consider the summation over terms $\alpha_m(x_{n-m}) \cdot f(x_{n-m})$, where f is also a neural network. We formalize this idea in Section 4.

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

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

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

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

where

- $F, S : \mathbb{R}^{d \times M} \to \mathbb{R}^{d_I \times M}$ are neural networks described below,
- $\boldsymbol{\sigma}$ is a normalized activation function independent on each row, i.e.

$$\boldsymbol{\sigma}((a_1^T, \ldots, a_{d_I}^T)^T) = (\sigma(a_1)^T, \ldots, \sigma(a_{d_I})^T)^T \tag{5}$$

  for any $a_1, \ldots, a_{d_I} \in \mathbb{R}^M$ and σ such that $\sigma(a)^T \mathbf{1}_M = 1$ 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 $F(\mathbf{x}_n^{-M})$. We are going to consider S to be a fully convolutional network (composed solely of convolutional layers) and F of the form

$$F(\mathbf{x}_n^{-M}) = W \otimes \left[\mathrm{off}(x_{n-m}) + x_{n-m}^I\right]_{m=1}^M \tag{6}$$

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

$$\hat{y}_n = \sum_{m=1}^{M} W_m \otimes (\mathrm{off}(x_{n-m}) + x_{n-m}^I) \otimes \sigma(S_m(\mathbf{x}_n^{-M})), \tag{7}$$

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 off and S respectively the *offset* and *significance* (sub)networks. The network scheme is shown in Figure 2. Note that when $\mathrm{off} \equiv 0$ 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 $\mathrm{off}(x_{n-m})$ 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.

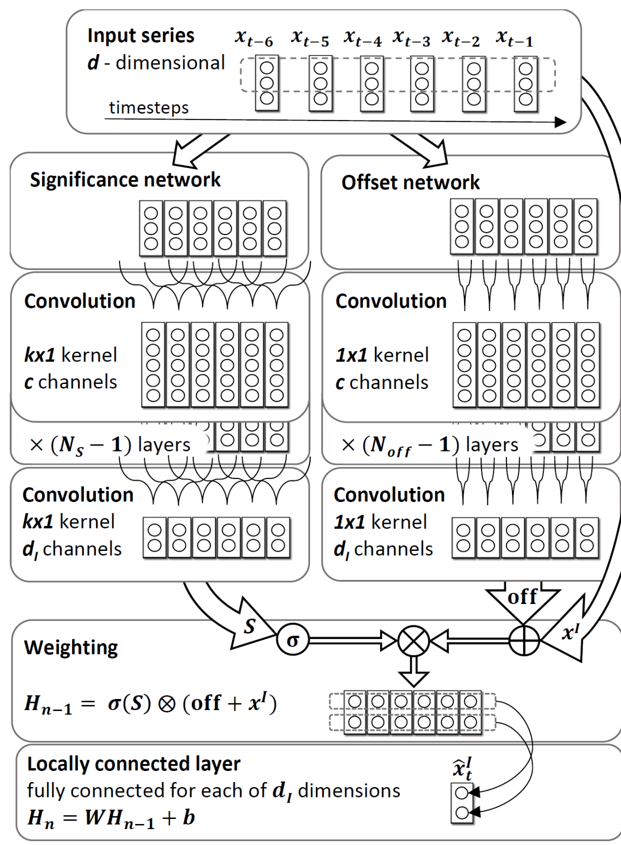


Figure 2. A scheme of the proposed SOCNN architecture. The network preserves the time-dimension up to the top layer, while the number of features per timestep (filters) in the hidden layers is custom. The last convolutional layer, however, has the number of filters equal to dimension of the output. The *Weighting* frame shows how outputs from offset and significance networks are combined in accordance with Eq. 7.

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[2], artificially generated *synchronous* and *asynchronous* datasets[3] and household *electricity* consumption dataset available from UCI repository (Lichman, 2013)[4][5]. 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 (Hochreiter & Schmidhuber, 1997) and 25-layer ResNet (He et al., 2015). 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[6]. 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.

[2]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.

[3]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.

[4]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 Repository website https://archive.ics.uci.edu/.

[5]The code for experiments and simulated series are available online at https://github.com/mbinkowski/nntimeseries.

[6]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 $\sigma(x) = \max(x, ax)$ (Maas et al., 2013) with leak rate of $a = .1$ was used in all layers except the top ones.

Results

Table 1 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 benchmarks. 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 1. 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.

model	VAR	CNN	ResNet	LSTM	SOCNN
Synchronous 16	0.841	0.152	**0.150**	0.152	0.154
Synchronous 64	0.364	**0.028**	**0.028**	**0.028**	0.029
Asynchronous 16	0.577	0.040	0.032	0.027	**0.017**
Asynchronous 64	0.318	0.041	0.046	0.050	**0.032**
Electricity	0.729	0.366	0.359	0.463	**0.158**
Quotes	1.000	0.975	3.565	3.696	**0.420**

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 $(\mathbf{x}_n^{-M}, y_n)$ we add noise of magnitude to every 5th of the past observations ($x'_{n-5k} = x_{n-5k} + \xi\epsilon$) and observe how the prediction errors change for each trained model, for varying ξ. Figure 3 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

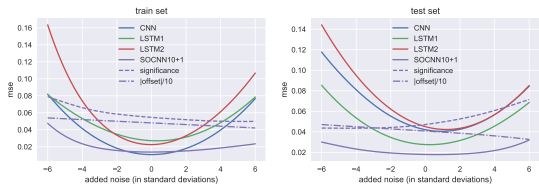


Figure 3. Changes in the prediction mean squared error with respect to the varying noise in 20% of input steps. LSTM1 and LSTM2 denote respectively one- and two-layer LSTMs. SOCNN appears to be more robust and adaptive to unseen data (note the uncentered curves for the other models for the test set), 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.

network architecture for time series prediction that proved successful in tested asynchronous regression tasks.

The proposed model can be further extended by adding intermediate weighting layers of the same type in the network structure. Another possible generalization that requires further empirical studies can be obtained by leaving the assumption of independent offset values for each past observation, i.e. considering not only 1x1 convolutional kernels in the offset sub-network.

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

Bengio, Yoshua, Ducharme, Réjean, Vincent, Pascal, and Janvin, Christian. A neural probabilistic language model. *J. Mach. Learn. Res.*, 3:1137–1155, March 2003. ISSN 1532-4435. URL http://portal.acm.org/citation.cfm?id=944966.

Borovykh, Anastasia, Bohte, Sander, and Oosterlee, Cornelis W. Conditional time series forecasting with convolutional neural networks, March 2017. URL http://arxiv.org/abs/1703.04691v1.pdf.

Bun, Joël, Bouchaud, Jean-Philippe, and Potters, Marc. Cleaning large correlation matrices: tools from random matrix theory. *Physics Reports*, 666:1–109, 2017.

Cho, Kyunghyun, Courville, Aaron, and Bengio, Yoshua. Describing multimedia content using attention-based Encoder–Decoder networks. *IEEE Transactions on Multimedia*, 17(11):1875–1886, July 2015. ISSN 1520-9210. doi: 10.1109/tmm.2015.2477044. URL http://dx.doi.org/10.1109/tmm.2015.2477044.

Chung, Junyoung, Gulcehre, Caglar, Cho, KyungHyun, and Bengio, Yoshua. Empirical evaluation of gated recurrent neural networks on sequence modeling, December 2014. URL http://arxiv.org/abs/1412.3555.

Collobert, Ronan and Weston, Jason. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pp. 160–167. ACM, 2008.

Cont, Rama. Empirical properties of asset returns: stylized facts and statistical issues. *Quantitative Finance*, 1(2): 223–236, 2001.

Cunningham, John P, Ghahramani, Zoubin, Rasmussen, Carl Edward, Lawrence, ND, and Girolami, M. Gaussian processes for time-marked time-series data. In *AISTATS*, volume 22, pp. 255–263, 2012.

Dauphin, Yann N., Fan, Angela, Auli, Michael, and Grangier, David. Language modeling with gated convolutional networks, December 2016. URL http://arxiv.org/abs/1612.08083.pdf.

Fama, Eugene F. Efficient capital markets: A review of theory and empirical work. *The journal of Finance*, 25 (2):383–417, 1970.

Gamboa, John Cristian Borges. Deep learning for time-series analysis. *arXiv preprint arXiv:1701.01887*, 2017.

Grave, Edouard, Joulin, Armand, Cissé, Moustapha, Grangier, David, and Jégou, Hervé. Efficient softmax approximation for GPUs, December 2016. URL http://arxiv.org/abs/1609.04309.pdf.

Hamilton, James Douglas. *Time series analysis*, volume 2. Princeton university press Princeton, 1994.

He, Kaiming, Zhang, Xiangyu, Ren, Shaoqing, and Sun, Jian. Deep residual learning for image recognition, December 2015. URL http://arxiv.org/abs/1512.03385.pdf.

Heaton, J. B., Polson, N. G., and Witte, J. H. Deep learning in finance, February 2016. URL http://arxiv.org/abs/1602.06561.pdf.

Hochreiter, Sepp and Schmidhuber, Jürgen. Long Short-Term memory. *Neural Computation*, 9(8):1735–1780, November 1997. ISSN 0899-7667. doi: 10.1162/

neco.1997.9.8.1735. URL http://dx.doi.org/10.1162/neco.1997.9.8.1735.

Hwang, Yunseong, Tong, Anh, and Choi, Jaesik. Automatic construction of nonparametric relational regression models for multiple time series. In *Proceedings of the 33rd International Conference on Machine Learning*, 2016.

Jozefowicz, Rafal, Vinyals, Oriol, Schuster, Mike, Shazeer, Noam, and Wu, Yonghui. Exploring the limits of language modeling, February 2016. URL http://arxiv.org/abs/1602.02410.pdf.

Krizhevsky, Alex, Sutskever, Ilya, and Hinton, Geoffrey E. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pp. 1097–1105, 2012.

Laloux, Laurent, Cizeau, Pierre, Potters, Marc, and Bouchaud, Jean-Philippe. Random matrix theory and financial correlations. *International Journal of Theoretical and Applied Finance*, 3(03):391–397, 2000.

Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, November 1998. ISSN 00189219. doi: 10.1109/5.726791. URL http://dx.doi.org/10.1109/5.726791.

Li, Steven Cheng-Xian and Marlin, Benjamin M. A scalable end-to-end gaussian process adapter for irregularly sampled time series classification. In *Advances in Neural Information Processing Systems*, pp. 1804–1812, 2016.

Lichman, M. UCI machine learning repository, 2013. URL http://archive.ics.uci.edu/ml.

Maas, Andrew L., Hannun, Awni Y., and Ng, Andrew Y. Rectifier nonlinearities improve neural network acoustic models. *JMLR: W&CP*, 28, 2013.

Mathieu, Michael, Couprie, Camille, and LeCun, Yann. Deep multi-scale video prediction beyond mean square error, February 2016. URL http://arxiv.org/abs/1511.05440.pdf.

McNelis, Paul D. *Neural networks in finance: gaining predictive edge in the market*. Academic Press, 2005.

Mozer, Michael C. Neural net architectures for temporal sequence processing. In *Santa Fe Institute Studies in the Sciences of Complexity*, volume 15, pp. 243–243, 1993.

Petelin, Dejan, Šindelář, Jan, Přikryl, Jan, and Kocijan, Juš. Financial modeling using gaussian process models. In *Intelligent Data Acquisition and Advanced Computing Systems (IDAACS), 2011 IEEE 6th International Conference on*, volume 2, pp. 672–677. IEEE, 2011.

Sak, Hasim, Senior, Andrew W, and Beaufays, Françoise. Long short-term memory recurrent neural network architectures for large scale acoustic modeling. In *Interspeech*, pp. 338–342, 2014.

Schmidhuber, Jürgen. Deep learning in neural networks: An overview. *Neural networks*, 61:85–117, 2015.

Sims, Christopher A. Money, income, and causality. *The American economic review*, 62(4):540–552, 1972.

Sims, Christopher A. Macroeconomics and reality. *Econometrica: Journal of the Econometric Society*, pp. 1–48, 1980.

Sirignano, Justin. Extended abstract: Neural networks for limit order books, February 2016. URL http://arxiv.org/abs/1601.01987.

Srivastava, Rupesh K., Greff, Klaus, and Schmidhuber, Jürgen. Highway networks, November 2015. URL http://arxiv.org/abs/1505.00387.

Tobar, Felipe, Bui, Thang D, and Turner, Richard E. Learning stationary time series using gaussian processes with nonparametric kernels. In *Advances in Neural Information Processing Systems*, pp. 3501–3509, 2015.

van den Oord, Aaron, Dieleman, Sander, Zen, Heiga, Simonyan, Karen, Vinyals, Oriol, Graves, Alex, Kalchbrenner, Nal, Senior, Andrew, and Kavukcuoglu, Koray. WaveNet: A generative model for raw audio, September 2016a. URL http://arxiv.org/abs/1609.03499.

van den Oord, Aaron, Kalchbrenner, Nal, Vinyals, Oriol, Espeholt, Lasse, Graves, Alex, and Kavukcuoglu, Koray. Conditional image generation with PixelCNN decoders, June 2016b. URL http://arxiv.org/abs/1606.05328.

Weissenborn, Dirk and Rocktäschel, Tim. MuFuRU: The Multi-Function recurrent unit, June 2016. URL http://arxiv.org/abs/1606.03002v1.pdf.