**Deep Learning of Heteroskedastic Volatility Models   
for Risk Estimation**

[*Nikolay Nikolaev*](file:///c:\HTML\index.html)

<https://nikolaevny.github.io/>

The volatility (changing variance) in time series of returns on assets impacts the pricing of financial instruments, and it is a key concept in automated algorithmic trading, option pricing, portfolio management and financial market regulation. The common assumption is that the return volatility is highly predictable. Popular tools for capturing the variance of the return distribution are the Generalized Autoregressive Conditional Heteroscedastic (GARCH) models [1], [2]. These models however are difficult to estimate [3] (typically by maximum likelihood optimization) and suffer from mistakes, which limits their practical usefulness. This short article explains that such difficulties can be mitigated by proper treatment of the temporal structures of the GARCH, and offers a contemporary approach to their calibration based on a deep learning technique.

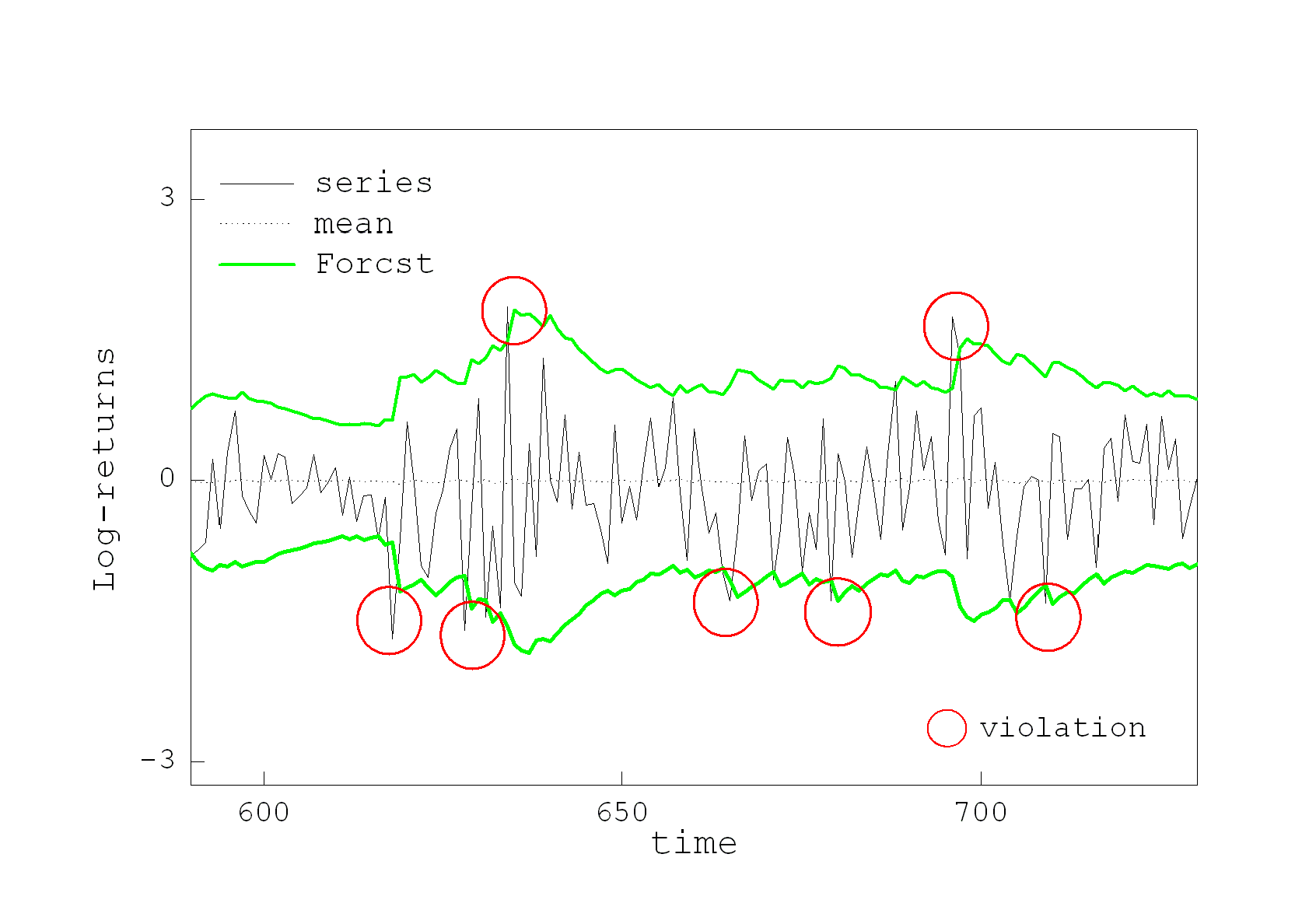
Alternative non-linear NGARCH models that are more reliable to estimate numerically have been designed using neural network formulations [4]. The neural nonlinear NGARCH extend the representation power of the original linear volatility models by non-linear terms, thus they yield more flexible descriptions. Although robust to misspecification errors, the NGARCH models did not become popular because the feed-forward neural networks cannot handle general spatial-temporal information beyond the selected inputs (that is, they impose a strong limit on the duration of the temporal events as they manipulate data through fixed time windows). Moreover the NGARCH parameters were trained with static algorithms that do not take into account the time as a parameter.

Further research proposed RGARCH models [5],[6] which are enhancements using recurrent (feedback) network architectures. The feedback networks provide adequate description of the inherently recursive GARCH models as they are driven by external inputs as well as by internal context (memory) signals that carry temporal information. The memory retains information from the past, and thus it helps to track better long-term behavioural patterns. The motivation to consider RGARCH for volatility modelling comes from the following advantages of recurrent networks: 1) they can infer time-varying return distributions as they explicitly describe heteroskedastic dependencies in serial data; 2) they allow accommodation of nonlinearities through different activation functions; and 3) they enable straightforward calibration with standard learning and optimization procedures.

A weakness in the above recurrent RGARCH approaches is that they still seek the model parameters with static training algorithms, that is they do not propagate the gradient information through time, therefore they do not use fully the temporal information. It should be emphasised that the main reason for using recurrent networks to develop GARCH models should be to exploit their property of being dynamic machines that can learn time-dependent functions. That is why, proper temporal derivatives for training recurrent variance networks were formulated recently to handle accurately the time relationships in the data [7]. These derivatives are actually nonlinear generalizations of the analytical derivatives available for linear variance models [8]. The proper dynamic derivatives were used in optimizers to evaluate dynamic DGARCH [9] models.

Our current research develops a deep learning algorithm [10] for dynamic heteroskedastic volatility models (D2GARCH) made as recurrent neural networks. The learning algorithm for D2GARCH creates a deep connectionist structure by unrolling the network in time. The unravelling is a process of transforming the temporal correlations into spatial relationships. The temporally dependent nodes and the recurrent connections are duplicated at each time instant, which leads to a deep topology of unfolded layers with shifted in time inputs trainable with static gradient-based algorithms. Then, the error is backpropagated through time (BPTT) in the sense that it is passed backward through the unrolled in time network, and the model parameters are adapted gradually to reflect the time ordering in the data sequence. Since the temporal error gradient decays and may vanish due to the unfolding (so it fails to contribute to the learning process), we apply the Kalman filtering technique [11] to diminish such effects and to speed up the convergence.

The usefulness of the D2GARCH variance network forecasts for managing risk exposure was analysed using the Value-at-Risk (VaR) measure [12],[13]. The VaR is a risk measure which tells us the maximum loss that may happen with a certain confidence over a given time period because of the fluctuations of market prices. The experimental methodology included bootstrapping the returns followed by VaR estimation. The obtained VaR predictions for 95% and 99% confidence levels were taken to compute likelihood ratio coverage tests as recommended [12]. When applied to a recent EUR/GBP exchange rates series it was found that D2GARCH shows similar risk performance to the other models (GARCH, NGARCH, RGARCH) on 99% VaR, and it is slightly better on 95% VaR estimation. The standard linear GARCH(1,1) showed unacceptably large failure rates on 99% and 95% VaR for the left tail, so it may be inadequate for predicting risk.

The figure below illustrates a segment of the log-returns from the considered EUR/GBP exchange rates series along with the bootstrapped 95% quantile, and some of the violations exceeding the return values (enclosed in circles).

References:

[1] Engle,R.F. (1982). *Autoregressive Conditional Heteroscedasticity with Estimates of the Variance of UK Inflation*,

Econometrica, vol.50, pp.987-1007.

[2] Bollerslev,T. (1986). *Generalized Autoregressive Conditional Heteroscedasticity*, Journal of Econometrics, vol.31, pp.307-327.

[3] Zumbach G. (2000) The Pitfalls in Fitting Garch(1,1) Processes. In: Dunis C.L. (eds) *Advances in Quantitative Asset Management. Studies in Computational Finance*, vol 1. Springer, Boston, MA.

[4] Donaldson,R.G. and Kamstra,M. (1997). An Artificial Neural Network – GARCH Model of International Stock Return Volatility, *Journal of Empirical Finance*, vol.4 (1) pp.17-46.

[5] Freisleben,B. and Ripper,K. (1997). Volatility Estimation with a Neural Network, in*: Proc. of the IEE/IAFE Computational Intelligence for Financial Engineering Conference* (CIFEr-97), IEEE Press, pp.177-181.

[6] Miazhynskaia,T., Dorffner,G. and Dockner,E.J. (2003). Risk Management Application of the Recurrent Mixture Density Network Models, in*: Proc. Int. Conf. on Artificial Neural Networks* ICANN-2003, LNCS-2714, Springer, Berlin, pp.589-596.

[7] Nikolaev,N., Tino,P. and Smirnov,E.N. (2011). Time-Dependent Series Variance Estimation via Recurrent Neural Networks, In: T.Honkela et al (Eds.) *Proc. Int. Conf. on Artificial Neural Networks*, ICANN-2011, Finland, LNCS-6971, Springer, pp.176-184.

[8] Fiorentini, Calzolari,G. and Panattoni,L. (1996). Analytical Derivatives and the Computation of GARCH Estimates, *Journal*

*of Applied Econometrics*, vol.11, pp.399-417.

[9] Nikolaev,N., Tino,P. and Smirnov,E.N. (2013). Time-Dependent Series Variance Learning with Recurrent Mixture Density Networks, *Neurocomputing*. vol.122, pp.501–512.

[10] Goodfellow,I., Bengio,Y. and Courville,A. (2016). *Deep Learning*, MIT Press.

[11] Haykin,S. (2001) *Kalman Filtering and Neural Networks and Learning Machines*, NY: Wiley-Intersience.

[12] Jorion,P. (1996). *Value at risk: The new benchmark for managing financial risk*. New York: The McGraw-Hill Co..

[13] Christoffersen,P. (2003). *Elements of financial risk management*. San Diego, CA: Academic Press.