Neural Network-Based Calibration of Stochastic Volatility Models

Xingzhu Zhu*
(Dated: Mar 29. 2020)

I. Introduction

It is known that implied volatility surface (IVS) derived based on Black-Scholes (BS) model exhibits volatility smiles and at-the-money (ATM) skewness, suggesting a contradiction to the flat surface assumed by BS. Corresponding diffusive stochastic volatility models like Bates [1] capturing the stochastic nature of volatility were developed and then an overwhelming shift to rough stochastic volatility models like rough Bergomi model [2] emerged, however, these models encountered a practicality bottleneck caused by computational challenge in model calibration [3]. With pioneer research of financial pricing model calibration with Neural Network (NN) [4], one recent research has demonstrated that the NN-based approach could potentially offer a more accurate, faster, and more robust calibration method for stochastic volatility models [5].

II. Model Calibration

In stochastic volatility models, *Calibration* refers to tuning parameters to let the model volatility surface fit with the IVS. Mathematically, suppose the model under construction is parametrized by a parameter vector θ, belonging to a set of parameters Θ. The derivative instrument in this model is parametrized by $\zeta \in Z' \subset Z$. Denote the market value of relevant derivative contract by V, with the observed market price by $V^*(\zeta)$ and model price $V(\zeta;\theta)$. The model price $V(\zeta;\theta)$ is usually derived from a system of SDEs. For instance, in the Heston model [6], $V(\zeta;\theta) = V(S,t,\nu;K,T)$, the option price at time t, is derived from

$$dS_t = rS_t dt + \sqrt{\nu_t} S_t dW_t^s, \tag{1a}$$

$$d\nu_t = \kappa(\bar{\nu} - \nu_t)dt + \gamma\sqrt{\nu_t} dW_t^\nu, \tag{1b}$$

$$dW_t^\nu dW_t^s = \rho_{x,\nu} dt, \tag{1c}$$

Calibration then is the procedure to find the optimal model parameter $\hat{\theta}^*$, minimizing the distance, D, between the observed market price and modeled price, together with a regularization term $\lambda R(V)$, i.e.,

$$\hat{\theta}^* = \arg\min_{\theta\in\Theta}\{D[V(\zeta;\theta), V^*(\zeta)]_{\zeta\in Z'} + \lambda R(V)\} \tag{2}$$

A common practice is to choose distance function D as weighted least squares function and take the regularization hyperparamerter $\lambda = 0$, i.e.,

$$\hat{\theta}^* = \arg\min_{\theta\in\Theta}\{\sum_{\zeta\in Z'} w_\zeta [V(\zeta;\theta) - V^*(\zeta)]^2\} \tag{3}$$

This optimization problem then can be solved by Levenverg-Marquart algorithm [7] efficiently if the Jacobian J of $V(\zeta;\theta)$ and the true implied volatility map can be expressed explicitly, which are generally not the case in stochastic volatility settings [3]. Numerical mathematical tools then offer a notable remedy [3]:
Step I: Approximate V by $\hat{V}$ numerically with Monte Carlo simulation;
Step II: Adopt finite-difference method to estimate J by $\hat{J}$.

However, in order to get an estimated pricing map $\hat{V}$, a lot of stochastic volatility models require massive simulations in Step I [3], which fails the real-time requirements of model calibration given the fast changes in the market.

Note that recent developed neural network learning pipelines fit with the model calibration task described above [3] and evidences have shown that NN-based approach would reduce the computing time of implied volatilities in high-dimensional diffusive stochastic volatility models [8], academia has regained interest in research in calibration of pricing models in recent years [3].

III. NN Architecture and NN-based Calibration

Generally, an NN is composed by neurons and layers. Each neuron contains three consecutive operations: summation of weighted inputs, additional summation of a bias, and finally compute outputs with an activation function [5]. A cluster of neurons is treated as one hidden layer and a pile of layers defines the whole architecture of one NN [5]. In mathematical form, a simplest multilayer perceptron, consisting of two layers, with input x and output y, is defined by [8]:

$$y(x;w,b) = \psi^{(2)}(\sum_j w_j^{(2)} z_j^{(1)} + b^{(2)}), \text{ where} \tag{4a}$$

$$z_j^{(1)} = \psi^{(1)}(\sum_i w_{ij}^{(1)} x_i + b_j^{(1)}) \tag{4b}$$

Here, w_i is unknown weight vector, b_i is unknown bias vector, and $\psi()$ is activation functions like Sigmoid. With target function $f(x)$ (also observations), the loss function is defined as

$$L(w,b;x,f) = \delta(f(x), y(x;w,b)) + \lambda R(y) \tag{5}$$

* Student of Quantitative Finance, Faculty of Business and Economics, The University of Hong Kong

Then the "learning" of parameters (w, b) is the optimization process of solving

$$(w^*, b^*) = \arg\min_{w,b} L(w, b; x, f) \tag{6}$$

In the non-NN stochastic volatility model calibration task, two numerical methods are involved. For example, to calibrate the implied volatility surface based on Heston, first we need to use the COS method for calculation of option prices modeled by Heston, and second the Brent's method is applied to find BS implied volatility [5]. Nonetheless, with the NN-based approach, training the mapping between the implied volatility and the Heston model parameters only needs once, and is off-line although time-consuming [5]. After this forward phase, on-line and computationally cheap calibrations can be achieved by feeding real-time market prices and implied volatilities into the NN trained in forward phase, i.e., solving the inverse mapping and finding the real-time Heston model parameters [5].

IV. Conclusion

Financial engineering is commonly termed to describe the construction process of new financial instruments using existing contracts in the market. The model designing and pricing process can be abstruse and heavily mathematical-driven. However, no matter how profound pricing models are, there is no single model that can perfectly fit the real market. Thus, asymptotic real-time calibration is of vital importance in the maintenance and management phase after the launch of a new product, which serves as a bridge between the implied volatility and the theoretical models. The IVS in some sense reflects the market participants' aggregate expectation of future price volatility of the underlying. Hence, through the model calibration, the pricing model can reflect such expectation in its parameters and hence better fit with the market expectations.

In the classical calibration process, various numerical methods are raised in both academia and industry, but the computational bottleneck has impeded the further developments of practical calibration. The recent explosion of machine learning research, together with supporting open-sourced packages, has given rise to a great amount of faster parameter fitting techniques like NN, enabling more accessible and pragmatic model calibrations. More importantly, the NN-based framework proposed by Liu [5] does not pose restrictions on target functions nor the gradients of related mapping models. Moreover, it is flexible in terms of the number of available market data and calibration parameters [5]. This generic new tool may help to obtain more new insights in the realm of pricing model calibration.

With more and more advanced techniques emerging from machine learning sphere, it is expected that the NN-based stochastic volatility model calibration could solve the expensive computational limitation, while still maintain a good balance with accuracy. In fact, the research of automatic network architecture search might ultimately automatize the manual parameter-tuning process at the time being, and hence further catalyze the applications of NN-based calibration in pricing models.

[1] D. S. Bates, Jumps and stochastic volatility: Exchange rate processes implicit in deutsche mark options, The Review of Financial Studies **9**, 69 (1996).

[2] C. Bayer, P. Friz, and J. Gatheral, Pricing under rough volatility, Quantitative Finance **16**, 887 (2016).

[3] C. Bayer, B. Horvath, A. Muguruza, B. Stemper, and M. Tomas, On deep calibration of (rough) stochastic volatility models, arXiv preprint arXiv:1908.08806 (2019).

[4] A. Hernandez, Model calibration with neural networks, Available at SSRN 2812140 (2016).

[5] S. Liu, A. Borovykh, L. A. Grzelak, and C. W. Oosterlee, A neural network-based framework for financial model calibration, Journal of Mathematics in Industry **9**, 9 (2019).

[6] S. L. Heston, A closed-form solution for options with stochastic volatility with applications to bond and currency options, The review of financial studies **6**, 327 (1993).

[7] D. W. Marquardt, An algorithm for least-squares estimation of nonlinear parameters, Journal of the society for Industrial and Applied Mathematics **11**, 431 (1963).

[8] S. Liu, C. W. Oosterlee, and S. M. Bohte, Pricing options and computing implied volatilities using neural networks, Risks **7**, 16 (2019).