ERASMUS UNIVERSITY ROTTERDAM

Erasmus School of Economics

Master Thesis Econometrics and Management Science

Deep calibration of SABR stochastic volatility

Author
Hugo Stuijt
433154

Supervisor
S.H.L.C.G. Vermeulen
Second assessor
prof. dr. M. van der Wel

15 July 2021

Abstract

We present a calibration method for the SABR stochastic volatility model based on machine learning techniques. The goal is to increase calibration accuracy with little decrease in calibration speed. Using an artificial neural network, we approximate a very accurate SABR option pricing map from model parameters to implied volatility. The neural network approximation is more accurate than the traditional formulas from Hagan et al. (2002), and more efficient than the computations from the original pricing map. In a numerical experiment, we find that the neural network can calibrate to an implied volatility smile with higher accuracy than the traditional formulas while being significantly faster than the original pricing map. However, we find that care must be taken when using the neural network with market data. By applying the methods to S&P500 options, we observe higher uncertainty in the SABR parameters calibrated by the neural network, creating instabilities in the estimated vega hedges. Nevertheless, the results show the potential for neural networks to improve SABR calibration speed and accuracy significantly.

Keywords: *SABR, Stochastic Volatility, Machine Learning, Neural Networks*

Contents

1 Introduction

For the widely used SABR stochastic volatility model, a speed and accuracy tradeoff needs to be made when calibrating to option market prices. This research approximates a highly accurate SABR pricing map with a neural network to make fast and accurate calibration possible for options with a wide range of strikes and maturities.

Given the SABR model parameters, option prices are often calculated using the analytical approximations derived in Hagan et al. (2002). This makes very fast calibration of SABR parameters to market prices possible. However, it is well known that the Hagan option price approximations perform poorly for long maturities and extreme strikes, resulting in inaccurate pricing and hedging. More accurate option price approximations under the SABR model are available in the literature, although computationally more intense. Therefore, these methods are difficult to use in practice because of long calibration times. To overcome the speed and accuracy tradeoff in SABR calibration, we use an artificial neural network to approximate a highly accurate SABR option pricing map. Using the trained neural network, SABR calibration is about as fast as the original Hagan et al. (2002) formulas while the accuracy of the computationally heavy pricing map is maintained.

The well-known Black-Scholes model (Black and Scholes, 1973) is almost 50 years after its introduction still one of the most popular models for pricing European options. Most often, option prices are quoted in terms of *implied volatility* (IV). Given an options market price, the IV is such that the Black-Scholes option price equals the market price. Calculating the IV and option price sensitivities (*greeks*) for hedging is relatively straightforward with the Black-Scholes model, contributing to its popularity. However, the underlying asset is assumed to follow a lognormal distribution with constant volatility. In reality, the IV differs across the option contracts' strikes and maturities, resulting in the implied volatility surface.

Modelling the implied volatility surface is important for hedging and pricing of both vanilla and exotic options. A range of volatility models has been developed to approximate the implied volatility smile (implied volatility in relation to the strike price) or surface. The accuracy of calibrating volatility models to market prices influences the quality of the hedges (Hagan et al., 2002). This could lead to unnecessary risk on option positions if inaccurate models are used. In a trading environment, calibration speed is of crucial importance as well. Especially in turbulent market conditions, volatility models need to be re-calibrated almost instantly to keep pricing accurate.

Local volatility and stochastic volatility models are the two classes of models used most often for handling the volatility smile and surface. Local volatility models (Dupire, 1994) can be calibrated to fit a set of European option prices perfectly (Stoep et al., 2014), however, they do consist of multiple flaws. As found in Hagan et al. (2002), such models do not capture the dynamics of volatility very well. As the price of the underlying increases, the models predict that the smile shifts to lower prices. Bad volatility dynamics could result in unstable delta and vega hedges, often performing worse than the standard Black-Scholes hedges (Hagan et al., 2002). To address the shortcomings of the Black-Scholes model and local volatility models, the SABR (stochastic α, β, ρ) model was introduced in Hagan et al. (2002).

The SABR model is a stochastic volatility model in which the forward and volatility processes are correlated. The model is widely used in the industry because the volatility dynamics are captured well, and calibration to market prices is relatively simple. In the original paper, an analytic pricing map is derived which maps the model parameters to the implied volatility. Because of the simplicity of the original formula, the SABR model is calibrated to a market smile very fast. However, it is well known that the Hagan et al. (2002) implied volatility approximation generally performs poor for longer maturities or extreme strikes; see for example Antonov et al. (2013) and Islah (2009).

Although it might still be possible to fit an incorrect IV approximation to a market smile accurately, it means that the calibrated SABR parameters are incorrect. Due to the nature of the calculation of the SABR greeks, risk would be estimated wrong with incorrect SABR parameters (Bartlett, 2006; Hagan and Lesniewski, 2017). Moreover, the dynamics of the volatility smile are not captured well (Hagan et al., 2002).

Multiple improvements or alternative methods have been proposed in the literature to overcome the IV approximation error of the original formula; see for example McGhee (2011) and Antonov et al. (2013). These SABR IV approximations are very accurate but relatively slow to compute. Since many function evaluations are expected in the SABR calibration process, calibration speed is expected to be slow. Therefore, the alternative improved implied volatility approximations are not directly usable in a trading environment.

Inspired by promising work from McGhee (2018), this research uses machine learning techniques to overcome the SABR calibration bottleneck imposed by computationally intense IV estimates. McGhee (2018) uses a relatively simple single-layer artificial neural network (ANN) to approximate the SABR implied volatility function. The approximated function is a very accurate integration scheme to overcome the original Hagan et al. (2002) flaws. The training of the

network is computationally heavy, however, it can be done beforehand. After the network has been trained, IV estimates can be made within milliseconds with similar accuracy as the original pricing map. Although the methods from McGhee (2018) are applied using relatively restricted parameters and the network is trained on a limited time to expiry range, the results show the potential for neural networks to improve SABR calibration speed and accuracy significantly.

Applying machine learning methods within finance is often difficult due to strict regulations and the 'black box' nature of deep learning models. However, using neural networks only to approximate the computationally heavy pricing map from model parameters to option price, the original model is still used and valid. This way, SABR calibration becomes a two-step procedure. First, the map from model parameters to implied volatility is trained and stored. Having obtained a trained network, model calibration is performed using traditional methods.

This research aims to create a fast and accurate SABR calibration procedure ready to use on short-and long-term options. Creating a very general calibration procedure means that training of the network can be done once, after which the methods can be applied to options on a wide range of asset classes. We build upon the neural network approach of McGhee (2018) by approximating the Antonov et al. (2013) SABR option pricing map with a neural network. The Antonov et al. (2013) implied volatility is more accurate than the traditional Hagan et al. (2002) approximations but slower to compute. We analyse the accuracy of the neural network approximations and whether this translates to better SABR calibration. The neural network is trained on a wide range of parameters and maturities of one week up to 10 years. We find that the neural network can closely approximate the accurate Antonov implied volatility while about as fast to compute as the traditional Hagan method.

Our main findings show that the neural network is more accurate in approximating implied volatilities and SABR greeks than the traditional Hagan formulas. The accuracy improvements are also found in calibration on simulated data, where the greeks are estimated better than the Hagan approximation. SABR calibration is performed over 200 times faster than the Antonov method it is based on, whereas it is only marginally slower than the already fast Hagan method.

We show that the methods can be applied directly to S&P500 option market data. However, the neural network vega estimates on market data are relatively unstable compared with the benchmarks. This is confirmed using a Bayesian calibration experiment performed to quantify the uncertainty in calibrated SABR parameters. Although the network performs very good on simulated data, we find higher parameter uncertainty for the neural network calibration on S&P500 option data.

2 Literature

After the introduction of the SABR model with the analytical implied volatility approximation in Hagan et al. (2002), multiple variants of the approximation have been developed. An improved version of the Hagan et al. (2002) asymptotic approximation was derived in Obłój (2008). This improved analytical method, however, still degrades in quality for large strikes and maturities (Lokvancic, 2020).

More elaborate extensions can be found in Islah (2009) and Antonov et al. (2013). The methods from Islah (2009) consist of three-dimensional integrals, which are slow to numerically approximate. The methods from Antonov et al. (2013) consist of two-dimensional integrals. However, a very accurate approximation is derived, which simplifies the problem to one-dimensional integrals. Note that the two methods calculate the price of a European call option, after which the implied volatility can be calculated via, e.g. Black-Scholes or Bachelier. This differs from the Hagan et al. (2002) formulas, which directly calculate the implied volatility. For the Black-Scholes implied volatility calculation, a root-finding algorithm is needed in addition to the numerical integration. In practice, this will not make a considerable contribution to the computational burden.

Alternatively, simulation methods have been proposed as a way of option pricing under the SABR model. Leitao et al. (2017b) proposes a one time-step Monte Carlo simulation method for accurate pricing, best applicable for pricing short-term options. Leitao et al. (2017a) proposes a multi-timestep generalisation, better suited for long-term options. Although relatively fast compared with conventional Monte Carlo methods, it would still be the bottleneck in SABR calibration due to the number of evaluations.

A low-bias Monte Carlo simulation scheme was proposed in Chen et al. (2012) which is more robust to unstable behaviour of the more general Euler discretisation scheme. This more general method is applied in Fernández et al. (2013). It is concluded that in their application the Chen et al. (2012) simulation scheme does not directly outperform conventional Monte Carlo methods. Fernández et al. (2013) calibrates the SABR model using regular Monte Carlo simulations but increases calibration speed by using multiple GPUs.

Multiple papers have been devoted to applying machine learning techniques to stochastic volatility models. In general, two main approaches can be considered, which we elaborate upon using terminology introduced in Bayer et al. (2019).

One-step approach

The goal of the one-step approach, introduced in Hernández (2016), is to use a neural network to approximate the complete calibration process directly. Thus, both the option price approximation and the parameter calibration are learned by the network. The concept can be applied to any stochastic volatility model. In the context of SABR, this means that a neural network is trained on a large number of IV smiles with SABR parameters as output. Having obtained a trained neural network, a market implied volatility smile is used as input to the trained network, and the approximated calibrated parameters will be the output.

Since SABR calibration is reduced to a single step in the one-step approach, model calibration is expected to be very fast. However, network training can be challenging. Hernández (2016) trains a neural network by calibrating SABR on historical market data, limiting the number of usable data points. Moreover, training the model using a better SABR IV approximation will be time-consuming, if not impossible. Finally, it is not guaranteed that the model performs good to unseen market data, as neural networks generally are not good at out of sample estimates.

Two-step approach

In the two-step approach, a neural network is used to approximate the map from model and contract parameters to option price (either the actual price or implied volatility). After that, the model can be calibrated using traditional methods but with a speed advantage in the pricing calculation. This process is therefore reverse to the one-step approach. The neural network takes the model parameters as input, as well as variables specifying the option contract. The approximated option price is the output of the network. A neural network is easier to train in the two-step approach since simulated data can be used with less effort, making the network perform better in unseen market conditions (Bayer et al., 2019).

In general, the two-step approach can be performed in two different ways as well. Following the terminology of Bayer et al. (2019), in the *pointwise* approach a neural network is trained using the model parameters and a single strike and maturity as input. This gives rise to a single option price as output. Alternatively, in the *gridwise* approach, the network is trained on a fixed grid of maturities and strikes. Instead of a single option price as output, the network outputs all option prices on which it has been trained. To obtain a price estimate which does not fall precisely on the grid, interpolation needs to be used. A comparison between the one-step and two-step methods, as well as pointwise and gridwise learning, can be found in Bayer et al. (2019). The authors find that the one-step approach to calibration is fastest on the cost of worse model interpretability and a higher variance in the neural network.

In Horvath et al. (2021), a general two-step gridwise learning approach is described. The methods, not limited to a specific volatility model, are applied to the Bergomi volatility model. First, a neural network is trained on a pricing map from model parameters to implied volatility. The neural network is trained offline, after which the approximated map is used online in the model calibration. Comparing this approach with Monte Carlo methods shows that the neural network calibration performs over 9,000 times faster while maintaining the accuracy of the original pricing model.

McGhee (2018) applies the two-step approach specifically to the SABR model. The network is trained using a fixed grid of strike prices. In contrast to Horvath et al. (2021), time to expiry is not included on the grid but as an additional variable. A computationally intense integration scheme is used as a pricing map, mapping the SABR parameters to European options implied volatility. This implied volatility approximation should be much more accurate than the original Hagan et al. (2002) formulas. The integration scheme is approximated using a single-layer artificial neural network, resulting in over 65 times faster calculation speeds while keeping the accuracy of the original pricing model. However, the performance of the model at calibrating to volatility smiles is not further explored. Although the implied volatility approximation in McGhee (2018) is limited to lognormal distributed prices ($\beta = 1$), it suggests that even using a relatively simple artificial neural network has the potential to significantly improve SABR calibration speed and accuracy. Lokvancic (2020) builds upon the results of McGhee (2018) by storing neural network results in a lookup table. Using this method, similar speed improvements as McGhee (2018) are achieved.

3 Data

The model performance to market data will be analysed on S&P500 options, obtained from WRDS OptionMetrics. The S&P500 index value (spot) is obtained from Bloomberg. We consider both weekly and monthly European style options from 1 January 2010 to 31 December 2020. We only keep options with nonzero volume for more accurate prices and consider strike prices between 40% and 160% of the index value. The Black-Scholes implied volatility is present in the database and is not required to be calculated. We also consider the best bid and best offer prices for each option. The S&P500 forward price for the corresponding option maturities is obtained from OptionMetrics as well. The final dataset for 2769 trading days consists of options with a maximum maturity between 21 and 36 months.

4 Methodology

The SABR model is used to parameterise an option markets implied volatility smile for a single maturity. The model is often used with the original formulas derived in Hagan et al. (2002). For given SABR parameters, the Hagan et al. (2002) formulas provide an option price in terms of Black-Scholes implied volatility or Bachelier implied volatility. Both implied volatilities will thus result in the same monetary value of the option. In this research, we always consider the Black-Scholes implied volatility due to it being most popular in the context of equity options.

The optimal SABR parameters are calibrated to market prices by minimizing the sum of squared errors of the SABR prices over the strikes. With the analytical Hagan et al. (2002) pricing formulas, implied volatilities are calculated very quick for a single smile. Therefore, calibration of SABR parameters to market data is very fast as well. However, the goal of this research is to use the more accurate SABR pricing function derived in Antonov et al. (2013). This method is computationally heavy, increasing the total calibration time of SABR parameters drastically.

To decrease the time to compute Antonov et al. (2013) implied volatilities, we approximate the function with an artificial neural network. Having obtained an approximation of an accurate SABR option pricing scheme, we analyse whether SABR calibration with the neural network is more accurate than the traditional Hagan et al. (2002) formulas. Furthermore, we examine whether improvements in calibration time have been made with respect to the Antonov implied volatility.

4.1 SABR calibration

In the SABR model derived in Hagan et al. (2002), the forward price $F(t,T)$ and instantaneous volatility $A(t)$ follow the following stochastic differential equations

$$\begin{aligned} dF(t,T) &= A(t)F(t,T)^{\beta}dW_1(t), & F(0,T) &= f, \\ dA(t) &= vA(t)dW_2(t), & A(0) &= \alpha, \\ dW_1(t)dW_2(t) &= \rho dt, \end{aligned} \tag{1}$$

where $\alpha, v > 0$, $-1 \leq \rho \leq 1$ and $0 \leq \beta \leq 1$. The standard Brownian motions W_1 and W_2 are correlated through ρ. v is the volatility of volatility parameter and α is the initial volatility. Finally, f is the initial forward price. Two special cases for the parameter β can be considered. When $\beta = 0$, we consider a stochastic normal model, and when $\beta = 1$ a stochastic lognormal model. For $0 < \beta < 1$, the forward price follows a CEV process, we refer to Cox (1996) for more

details.

It is common practice to fix the exponent β beforehand since β and ρ affect the volatility smile in a similar way (Hagan et al., 2002). Therefore, problems can occur when calibrating the model to market prices. Many heuristics are available to determine the value of β beforehand, see for example Zhang and Fabozzi (2016). Often, it is updated relatively infrequently and only if there is a significant change in market conditions (Lokvancic, 2020). With interest rate derivatives, for example, $\beta = 1$ is often used in a high interest rate environment and $\beta = 0$ in a low interest rate environment (Zhang and Fabozzi, 2016). For the remainder of this paper, we fix the parameter to $\beta = 0.5$ following Stoep et al. (2014). The initial forward price f is fixed as well, most often given by a market price. This leaves the initial volatility α, correlation ρ and volatility of volatility v to be calibrated.

Denote the SABR implied volatility approximation for expiration T_j and strike K_i as $\hat{\sigma}_{IV}(\zeta_{ij}, \Theta)$ and the corresponding observed market implied volatility as $\sigma_{mkt}(\zeta_{ij})$. Here we have $\zeta_{ij} = \{K_i, T_j, f_j\}$ the option contract parameters and $\Theta = \{\alpha, \rho, v\}$ the SABR parameters with fixed β. To calibrate the SABR model to a market smile corresponding to maturity T_j we minimize the sum of squared errors

$$\arg\min_{\Theta} \sum_{i} \Big(\sigma_{mkt}(\zeta_{ij}) - \hat{\sigma}_{IV}(\zeta_{ij}, \Theta)\Big)^2. \tag{2}$$

To solve Equation (2) as fast as possible, we follow the efficient calibration procedure from Le Floc'h and Kennedy (2014). We solve the problem as a nonlinear least squares regression using the Levenberg-Marquardt algorithm (Levenberg, 1944; Marquardt, 1963). Le Floc'h and Kennedy (2014) provide an approach to find starting values for the SABR parameters which are close to optimal, speeding up the calibration process.

4.2 Implied volatility approximation

To calibrate the SABR model parameters to observed market prices using Equation (2) we need the SABR implied volatility $\hat{\sigma}_{IV}$. The original Hagan et al. (2002) implied volatility approximation for $\zeta_{ij} = \{K_i, T_j, f_j\}$ and $\Theta = \{\alpha, \rho, v\}$ is given by

$$\hat{\sigma}_{IV}^{\text{Hagan}}(\zeta_{ij}, \Theta) = I^0(\zeta_{ij}, \Theta)\left(1 + I^1(\zeta_{ij}, \Theta)T_j\right), \tag{3}$$

where

$$I^0(\zeta_{ij},\Theta) = \frac{\alpha z_{ij}}{x(z_{ij})(f_jK_i)^{(1-\beta)/2}\left(1+\frac{(1-\beta)^2}{24}\log^2\left(\frac{f_j}{K_i}\right)+\frac{(1-\beta)^4}{1920}+\log^4\left(\frac{f_j}{K_i}\right)\right)},$$

$$I^1(\zeta_{ij},\Theta) = \frac{(1-\beta)^2}{24}\frac{\alpha^2}{(f_jK_i)^{1-\beta}}+\frac{1}{4}\frac{\rho\beta v\alpha}{(f_jK_i)^{(1-\beta)/2}}+\frac{2-3\rho^2}{24}v^2,$$

$$z_{ij} = \frac{v}{\alpha}(f_jK_i)^{(1-\beta)/2}\log\left(\frac{f_j}{K_i}\right),$$

$$x(z_{ij}) = \log\left(\frac{\sqrt{1-2\rho z_{ij}+z_{ij}^2}+z_{ij}-\rho}{1-\rho}\right).$$

In the at-money-case $f_j = K_i$ we have

$$I^0(\zeta_{ij},\Theta) = \frac{\alpha}{f_j^{1-\beta}}.$$

Because of its poor performance for longer maturities and extreme strikes, the Hagan approximation from Equation (3) will only serve as a benchmark. We use the methods from Antonov et al. (2013) instead, which provide a more accurate implied volatility approximation $\hat{\sigma}_{IV}^{\text{Antonov}}(\zeta_{ij},\Theta)$. Antonov et al. (2013) derives the forward price of a European call option denoted as $\hat{C}^{\text{Antonov}}(\zeta_{ij},\Theta)$. This process is extensive, and it is further described in Appendix A. However, it is important to note that the calculation of $\hat{C}^{\text{Antonov}}$ depends on numerical integration, making it relatively slow to compute.

Where the original Hagan et al. (2002) formula (3) directly returns the implied volatility, Antonov et al. (2013) returns the approximated European call option forward price $\hat{C}^{\text{Antonov}}$. After discounting the forward call price, we obtain the implied volatility approximation $\hat{\sigma}_{IV}^{\text{Antonov}}(\zeta_{ij},\Theta)$ from the Black-Scholes equation via Jäckel (2015). Since we are working with forward prices in the Black-Scholes equation, the forward call option price does not depend on the interest rate anymore.[1] The interest rate can thus be set to zero, reducing the input dimensionality of the Antonov et al. (2013) approximation.

4.3 Artificial neural network

To calibrate SABR parameters using $\hat{\sigma}_{IV}^{\text{Antonov}}$ in Equation (2), the numerical integration in $\hat{C}^{\text{Antonov}}(\zeta_{ij},\Theta)$ becomes the bottleneck for fast calibration. Although a single approximation

[1] The Black-Scholes price for a T-forward call option with strike K on T-forward price $F(t,T)$ at time t maturing at T is $C_F(t,T) = F(t,T)\Phi(d_1) - K\Phi\big(d_1-\sigma_t\sqrt{T-t}\big)$ with $d_1 = \frac{\log\left(\frac{F(t,T)}{K}\right)-\frac{1}{2}\sigma_t^2(T-t)}{\sigma_t\sqrt{T-t}}$ since $S_t = e^{-r(T-t)}F(t,T)$. Here, S_t is the underlying spot price at time t and σ_t is the volatility of the underlying at time t. See also Black (1976).

can be made relatively quick, many function evaluations are expected in the minimisation of Equation (2). To increase the calibration speed, we approximate the mapping from SABR and derivative parameters to Antonov et al. (2013) implied volatility using an artificial neural network. This will give rise to the implied volatility approximation $\hat{\sigma}_{IV}^{\text{ANN}}$. Like the Hagan and Antonov implied volatility approximations, the neural network approximation $\hat{\sigma}_{IV}^{\text{ANN}}$ can be used in the minimization of Equation (2) directly for SABR calibration.

In short, the goal of a neural network is to approximate a function $y = g^*(\boldsymbol{x})$ which maps the input $\boldsymbol{x}$ to the value y. A neural network now approximates this mapping with $y \approx g(\boldsymbol{x}, \boldsymbol{\theta})$. We speak of a 'network' since g in general consists of multiple functions chained together. For example, a neural network with n layers has the form $g(\boldsymbol{x}, \boldsymbol{\theta}) = g^{(n)}(g^{(n-1)}(...g^{(1)}(\boldsymbol{x}, \boldsymbol{\theta})...))$ where the optimal values for the weights $\boldsymbol{\theta}$ are determined in the training of the network.

The architecture of a feed-forward neural network now consists of one or more layers with multiple neurons. The network's input is transformed using weighted sums and (often) non-linear activation functions in the neurons. An important result regarding neural networks is the Universal Approximation Theorem (Cybenko, 1989). In short, the theorem states that a neural network with a linear output layer and at least one hidden layer can approximate any continuous function with any desired non-zero amount of error. This is provided a sigmoid-like activation function is used in the hidden layers, and enough neurons are used. The theorem has been expanded to a wider range of activation functions as well (Leshno et al., 1993). We refer to Hastie et al. (2001) and Goodfellow et al. (2016) for the basics of machine learning and neural networks.

In this research, we are interested in training an artificial neural network on the mapping $g^* : (\zeta, \Theta) \mapsto \hat{\sigma}_{IV}^{\text{Antonov}}(\zeta, \Theta)$. We choose to apply the aforementioned pointwise learning approach. This is in contrast to research of McGhee (2018) and Bayer et al. (2019) which uses the gridwise approach. We motivate the downside of using the gridwise approach as follows. Suppose we train a neural network on a fixed grid of 10 equally spaced strikes $K_1, K_2, ..., K_{10}$. After input of the SABR parameters and the maturity, the network outputs IV estimates $\hat{\sigma}_1, \hat{\sigma}_2, ..., \hat{\sigma}_{10}$. Suppose we are interested in the implied volatility for a strike that does not fall on the equally spaced grid. We would have to interpolate between the two strikes K_i and K_{i+1} which fall around the strike of interest. When the implied volatility smile contains a heavy curve, interpolation can result in an unsatisfactory fit at the (arguably most important) strikes around at the money. The described phenomenon is plotted in Figure 12 in Appendix C. These problems are solved by adding strikes to the grid around at the money until the desired accuracy is reached

or by reverting to the pointwise approach. In the latter case, the strike price is now added as an input to the network. This results in a single IV approximation in the output layer of the network.

4.3.1 Network architecture

Following the Universal Approximation Theorem, a single layer neural network could be sufficient for an adequate implied volatility approximation. This is also demonstrated by McGhee (2018) which uses a single layer neural network with the number of nodes ranging from 250 to 1,000. However, we choose a deep feed-forward network with four hidden layers and 30 nodes in each layer, following Bayer et al. (2019). The results of Eldan and Shamir (2016) motivate using networks with more layers and fewer nodes instead of fewer layers with more nodes. Horvath et al. (2021) indicate that the use of more than four layers does not significantly improve network performance. Moreover, the use of too many layers increases the risk of overfitting due to model complexity Goodfellow et al. (2016).

For the output layer, we take a linear activation function in line with the Universal Approximation Theorem. Many types of nonlinear activation functions are available for the hidden layers, all with their advantages and disadvantages. A common activation function used in the hidden layers is the rectified linear unit (ReLU) function given by $g(z) = \max\{0, z\}$. This function, introduced in Glorot et al. (2011), is simple, which could reduce training time. However, if $z < 0$ the gradient is equal to zero, which could impose problems during network training (also known as the 'vanishing gradient problem').

The 'softplus' function $g(z) = \ln(1+e^z)$ is a smooth version of the ReLU. However, use of it is generally discouraged in the hidden layers because of poor empirical performance (Goodfellow et al., 2016). Instead, following Horvath et al. (2021), we use the exponential linear unit (ELU) proposed in Clevert et al. (2016). This activation function is given by

$$g(z) = \begin{cases} \kappa(e^z - 1) & \text{if } z < 0 \\ z & \text{if } z \geq 0 \end{cases}, \tag{4}$$

where we set $\kappa = 1$. Although the vanishing gradient problem is not fully solved with the ELU activation function, the strictly positive gradient of (4) should make gradient-based learning of the network perform better (Clevert et al., 2016).

Since the goal is to map the SABR and option contract parameters to an implied volatility estimate, we have input nodes for the SABR parameters α, ρ and v. To decrease the dimension

of the input parameters, we standardise the strike price by dividing it with the initial forward price f such that we can set $f = 1$ in the Antonov et al. (2013) approximation. Therefore, we have additional nodes for strike K and time to maturity T, resulting in 5 input nodes. We have a single output node for the neural network implied volatility, denoted as $\hat{\sigma}_{IV}^{\text{ANN}}$. A schematic of the network architecture can be found in Figure 13, Appendix C.

4.3.2 Network training

To train the network on a large sample of input-output pairs, we use the minibatch gradient descent method with the Adam algorithm as optimiser (Kingma and Ba, 2017; Goodfellow et al., 2016). The training data is split into minibatches and using back-propagation, the gradient is computed on a single minibatch. A larger minibatch is a better approximation for the data but uses more memory and CPU power. However, the total training time for larger batch sizes is shorter compared with smaller batch sizes. Although it thus seems that a larger batch size is always better, training of the network is more likely to be stuck in local minima. Better model performance for smaller batch sizes is also observed in Keskar et al. (2017). Here, a large batch size is considered to be 10% of the training data size, and out-of-sample performance decreases with batch size. Therefore, we start with a small batch size and increase it until network training takes too much computing resources. At the same time, we make sure that the batch size does not come close to 10% of our training data size. We use a mean squared error loss function. We choose the number of epochs to be 200 and stop the training if the loss function on the validation set has not improved for 25 epochs.

We train the network on a wide range of parameter values and maturities. We create a training dataset using a fixed grid of parameters as well as randomly sampled parameters. The validation set consists out of randomly sampled parameters. For each sampled parameter set we calculate the Antonov implied volatility $\hat{\sigma}_{IV}^{\text{Antonov}}$. To make sure the network is only trained on realistic parameter values we fix the range of each parameter beforehand. For the time to expiry, we consider the following set of 17 maturities:

$$T \in \{7\text{d}, 14\text{d}, 30\text{d}, 60\text{d}, 90\text{d}, 180\text{d}, 270\text{d}, 1\text{y}, 2\text{y}, ..., 10\text{y}\}$$

This way, there is more information for shorter maturities available. For the standardized strike price K we consider the set of 13 strikes $K \in \{0.40, 0.50, ..., 1.60\}$. For α, ρ and v, we base the parameter ranges loosely on historical S&P500 data and make sure that historical values fall easily in our ranges. This way, possible extreme future market conditions can still be estimated

with the current model. In Appendix C Figure 14, historical values for α, ρ and v are shown. For each date, the S&P500 IV smile closest to 14 days to expiry and closest to 180 days to expiry has been calibrated using the traditional Hagan formula from Equation (3).

For initial volatility α, we fix the parameter range between 0 and 1. The parameter range for ρ is set between -1 and 0.5. For volatility of volatility parameter v, it seems less appropriate to fix the parameter range for all maturities. Especially for short-term options, v can get as high as 8 (or 800%). This is in contrast to longer-term options where v does not get higher than 2.2. We apply the heuristic introduced in McGhee (2018) to set upper and lower boundaries on v depending on the time to maturity T. We use

$$v_t = v_s \times \sqrt{\frac{T_s}{T_t}}, \tag{5}$$

such that $v_t < v_s$ if $T_t > T_s$. For $T_s = 30$ days we set the lower and upper bound for v_s to respectively .05 and 7. Fixing these bounds, the volatility of volatility of an option maturing in 10 years is bounded between 0.0045 and 0.64.

We create the fixed grid training dataset by dividing the fixed ranges of α, ρ and v into 20 equidistant points. Together with the ranges for maturity T and strike K, we have a total of $17 \times 13 \times 20^3 \approx 1.8$ million training samples. We add a further 2.2 million uniform randomly sampled points to the training set such that the total training set consists of around four million samples. The validation set consists out of two million uniform randomly sampled points. For some parameter combinations, a valid call price does not exist. Therefore, any invalid IV approximation is removed from the training and validation dataset. Before training, we scale the input and output data to have zero mean and standard deviation of one. The neural network has been implemented using Keras 2.4.3.

4.4 Evaluation

Evaluation of the proposed methods will be split into four main parts. First, we evaluate the accuracy of the neural network approximation. Next, we perform a numerical experiment to examine the speed and accuracy of the neural network in a controlled environment. We compare the neural network approach with the Hagan and Antonov implied volatility approximations. To verify practical applicability, we apply the methods to S&P500 options. Finally, we perform a Bayesian calibration experiment on both simulated data and S&P500 option data to give insight into the uncertainty of the calibrated SABR parameters.

Neural network accuracy

After training, it is necessary to verify that the neural network correctly approximates the Antonov et al. (2013) implied volatility. We do this by comparing the relative error of the neural network IV approximation with the Antonov et al. (2013) value over the entire strike/maturity surface. We define the relative error as

$$E_R(\hat{\theta}) = \frac{|\hat{\theta} - \theta|}{|\theta|}, \tag{6}$$

where $\hat{\theta}$ is the estimated value of parameter θ. To examine the neural network accuracy we thus calculate $E_R(\hat{\sigma}_{IV}^{\text{ANN}})$ with $\theta = \hat{\sigma}_{IV}^{\text{Antonov}}$. The relative error seems best suited for comparing the error in implied volatility estimates because the IV in general takes much higher values for smaller maturities as compared to higher maturities.

Numerical experiment

We perform a numerical analysis and compare the neural network implied volatility approximation with the Antonov et al. (2013) and Hagan et al. (2002) approximations. We are especially interested in the difference in calibration speed compared with the Antonov approximation, and the difference in accuracy compared with the Hagan approximation. To compare the calibration methods, we perform Monte Carlo simulations by using a discrete-time approximation for the stochastic processes in Equation (1). Using the Euler scheme, this now takes the following form:

$$\begin{aligned} \hat{F}(t_{i+1}, T) &= \hat{F}(t_i, T) + \hat{A}(t_i)\hat{F}(t_i, T)^{\beta}\Delta W_1(t_{i+1}), & \hat{F}(0, T) &= f, \\ \hat{A}(t_{i+1}) &= \hat{A}(t_i) + v\hat{A}(t_i)\Delta W_2(t_{i+1}), & \hat{A}(0) &= \alpha. \end{aligned}$$

The correlated increments from the Brownian motions W_1 and W_2 are approximated using

$$\begin{aligned} \Delta W_1(t_{i+1}) &= \sqrt{\Delta t} Z(t_{i+1}), \\ \Delta W_2(t_{i+1}) &= \sqrt{\Delta t}\Big(\rho Z(t_{i+1}) + \sqrt{1-\rho^2} Y(t_{i+1})\Big), \end{aligned}$$

where $Z(t_{i+1}), Y(t_{i+1}) \sim NID(0, 1)$ and $\Delta t = t_{i+1} - t_i$. We perform all simulations with 250,000 repetitions and 50 time steps such that $\Delta t = T/50$. Having obtained simulated forward prices, calculation of the implied volatility is trivial.

Instead of only comparing the methods based on implied volatility, we will also consider the SABR greeks. Following Hagan and Lesniewski (2017), we calculate the delta hedge (denoted

with Δ) and vega hedge (denoted with $\mathcal{V}$) as follows

$$\Delta = \Delta_{BS} + \mathcal{V}_{BS}\frac{\partial \hat{\sigma}_{IV}}{\partial f}, \tag{7}$$

$$\mathcal{V} = \mathcal{V}_{BS}\frac{\partial \hat{\sigma}_{IV}}{\partial \alpha}, \tag{8}$$

where Δ_{BS} and $\mathcal{V}_{BS}$ are the Black-Scholes delta and vega terms respectively. We compute the partial derivatives of $\hat{\sigma}_{IV}$ to α and f using central differences on $\hat{\sigma}_{IV}^{\text{Hagan}}, \hat{\sigma}_{IV}^{\text{Antonov}}$ or $\hat{\sigma}_{IV}^{\text{ANN}}$ where applicable. For the Monte Carlo simulated greeks, we thus also need to simulate the implied volatility around $f = 1$ and around the considered value for α to be able to use central differences. We note that it is not possible to compute $\frac{\partial \hat{\sigma}_{IV}}{\partial f}$ for $\hat{\sigma}_{IV}^{\text{ANN}}$ directly because the neural network is only defined in $f = 1$. We describe the solution for this in Appendix B.

Calculating accurate hedges is important for practical reasons, but it also shows that the volatility smile dynamics are captured well. In the calculation of the delta and vega hedges in Equations (7) and (8) the dynamics of the smile as the forward price and the initial volatility α move are entered through the partial derivatives $\frac{\partial \hat{\sigma}_{IV}}{\partial f}$ and $\frac{\partial \hat{\sigma}_{IV}}{\partial \alpha}$ respectively.

We perform the numerical experiment by creating a controlled environment with Monte Carlo simulations. We randomly generate 10,000 parameter sets consisting of SABR parameters α, ρ, v and time to maturity T. All parameters satisfy the bounds defined in Section 4.3.2, and we only consider the 25 strikes $0.40, 0.45, ..., 1.55, 1.60$. For each of the 10,000 parameter sets, we simulate the implied volatility smile and the greeks.

First, we treat the random SABR parameters as given and assume the Monte Carlo implied volatility and greeks as correct. With the neural network, Hagan and Antonov approximations, we calculate the implied volatility, delta and vega using the given parameters. For the implied volatility and greeks, we calculate the relative error as defined in Equation (6). When we denote the Monte Carlo implied volatility as $\hat{\sigma}_{IV}^{\text{MC}}$ we thus calculate $E_R(\hat{\sigma}_{IV}^{\text{ANN}})$ with $\theta = \hat{\sigma}_{IV}^{\text{MC}}$ for the neural network relative implied volatility error. We also compare the time to calculate a single implied volatility smile for all considered methods, which can be considered as the time it takes for a single function evaluation in the minimisation of Equation (2).

Next, we consider the random SABR parameters to be unknown and calibrate all methods on the 10,000 Monte Carlo implied volatility smiles. We will not look into the relative errors of the calibrated parameters themselves because these are difficult to interpret. Instead, using the calibrated values for α, ρ and v, we calculate the corresponding delta and vega hedges to obtain the relative errors of the calibrated hedges. This way, the quality of the calibration can

be compared without exploring differences in calibrated parameters. The calibration speed of all considered methods can now be compared as well.

S&P500 calibration

To examine the practical applicability of the neural network, we apply the methods on S&P500 option data. Note that we use the option strike price relative to the S&P500 index value such that we can set $f = 1$.[2] We assess the difference in SABR calibration time by calibrating all methods to a complete implied volatility surface. We consider all S&P500 options on 5 August 2019 maturing in one week or more and calibrate each volatility smile using the neural network, Antonov and Hagan approximations.

Furthermore, we calibrate the SABR model on options maturing in 30 days from 1 January 2010 to 31 December 2020 using the neural network approximation and the Hagan et al. (2002) approximation. Aside from evaluating the difference in calibrated parameters, we calculate and compare the at the money delta and vega hedges from Equations (7) and (8).

Bayesian calibration

Inspired by the methods from Bayer et al. (2019) and Duembgen and Rogers (2014) we perform a Bayesian calibration procedure. The goal is to quantify the uncertainty of the SABR calibration process using neural networks compared to the traditional Hagan et al. (2002) procedure. We treat the SABR model parameters $\Theta = \{\alpha, \rho, v\}$ as random variables and perform a nonlinear Bayesian regression. This way, we update prior beliefs on the SABR parameters with the likelihood of observing a given implied volatility smile.

Denote the volatility smile for maturity T_j as

$$\boldsymbol{y} = \big[y_j(K_1), ..., y_j(K_N)\big]',$$

such that $y_j(K_i)$ is the implied volatility for the strike K_i option with maturity T_j. Then, we introduce the SABR approximated implied volatilities as

$$\hat{\boldsymbol{F}}(\Theta) = \big[\hat{\sigma}_{IV}(\zeta_{1j}, \Theta), ..., \hat{\sigma}_{IV}(\zeta_{Nj}, \Theta)\big]'.$$

[2] We note that it is possible to rewrite a SABR model with initial volatility α and initial forward price f to an equivalent SABR model with a different initial forward price f'. We do this using a correction to the initial volatility. The proof of this is part of the derivations in Appendix B.

The regression now takes the form

$$\boldsymbol{y} = \hat{\boldsymbol{F}}(\Theta) + \boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon} \sim N\big(\mathbf{0}, \boldsymbol{\Sigma}\big). \tag{9}$$

Specifying $\boldsymbol{\Sigma}$ is a nontrivial task but an important issue for Bayesian calibration, as argued in Duembgen and Rogers (2014). Assuming $\boldsymbol{\Sigma}$ to be stochastic as well and using a flat prior specification would defeat the purpose of the Bayesian calibration, as the calibration would now revert to the original calibration problem without any uncertainty in the parameters. Choosing homoscedastic error terms seems too simplistic, especially for short term options. For deep out the money call options expiring one week from now, we would not expect much uncertainty in the volatility smile since the probability of expiring in the money is very low. This contrasts with deep in the money options, which move almost one-to-one with the underlying price, thus creating more uncertainty in the in the money part of the IV smile. For long term options, homoscedastic error terms might be more reasonable.

Alternatively, one could add noise to the smile itself. However, this process is difficult since a smile must be convex due to the no-arbitrage argument and independent noise is therefore not likely (Duembgen and Rogers, 2014). For this reason, we follow Bayer et al. (2019) and assume heteroscedastic residuals such that $\boldsymbol{\Sigma} = \text{diag}(\sigma_1^2, ..., \sigma_N^2)$. All together, the joint likelihood is assumed to be to be multivariate normal, formally $\boldsymbol{y} \mid \Theta \sim N(\hat{\boldsymbol{F}}(\Theta), \boldsymbol{\Sigma})$.

We are interested in the posterior distribution of the SABR parameters $p(\Theta|\boldsymbol{y}) \propto p(\boldsymbol{y}|\Theta)p(\Theta)$. We use a flat prior specification for $p(\Theta)$. Using Markov chain Monte Carlo methods from Foreman-Mackey et al. (2013) we approximate the posterior distribution numerically, sampling 2,500 SABR parameter pairs from the posterior.

We simulate an implied volatility smile with known SABR parameters and perform the Bayesian calibration procedure with the neural network and Hagan approximations. For the variances in $\boldsymbol{\Sigma}$, we take the variance of the Monte Carlo option prices over each strike. We then compare the marginal posterior distributions of the SABR parameters with the real parameter values. For each pair of SABR parameters sampled from the posterior distribution, we calculate the corresponding at the money delta and vega hedges using Equations (7) and (8). The distribution of the hedges with respect to the Monte Carlo hedges can now be analysed as well.

The same process is performed on an S&P500 implied volatility smile. As a proxy for the variances in $\boldsymbol{\Sigma}$ we take the bid-ask spread. We scale the bid-ask spread to be of the same order of magnitude as the variances of an equivalent smile simulated with Monte Carlo.

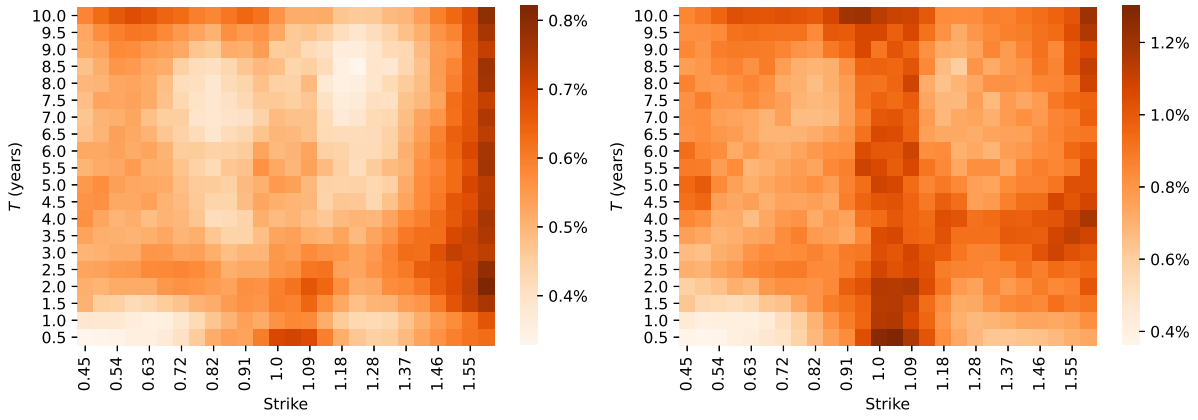
5 Results

5.1 Neural network accuracy

We train the network using the described methods and settle with a batch size of 2048. A plot of the mean squared error cost function for each epoch in the network training can be found in Figure 15 in Appendix C. Training of the network has stopped after 188 epochs since the cost function of the validation set has not decreased for 25 epochs at that point.

We evaluate the network accuracy by sampling random data. We generate five million random parameter combinations for SABR parameters α, ρ, v, strike K and maturity T. We sample uniformly from the parameter ranges defined in Section 4.3.2 while making sure that the range v is sampled from satisfies the bounds defined in Equation (5). For each parameter set, we calculate the Antonov et al. (2013) implied volatility. This process is thus identical to the creation of random training/validation data. This data, however, has never been used in neural network training. Next, we calculate the relative error of the neural network IV approximation compared with the Antonov implied volatility.

Figure 1 shows the mean and standard deviation of the relative implied volatility error of the neural network. Figure 1 (a) shows that the neural network manages to approximate the Antonov et al. (2013) implied volatility with good accuracy. The average relative error of the approximation is at most 0.82%. In general, the approximation degrades for larger strike prices. We also observe an increase in relative errors with time to maturity, although to a lesser extent.



(a) Average relative error

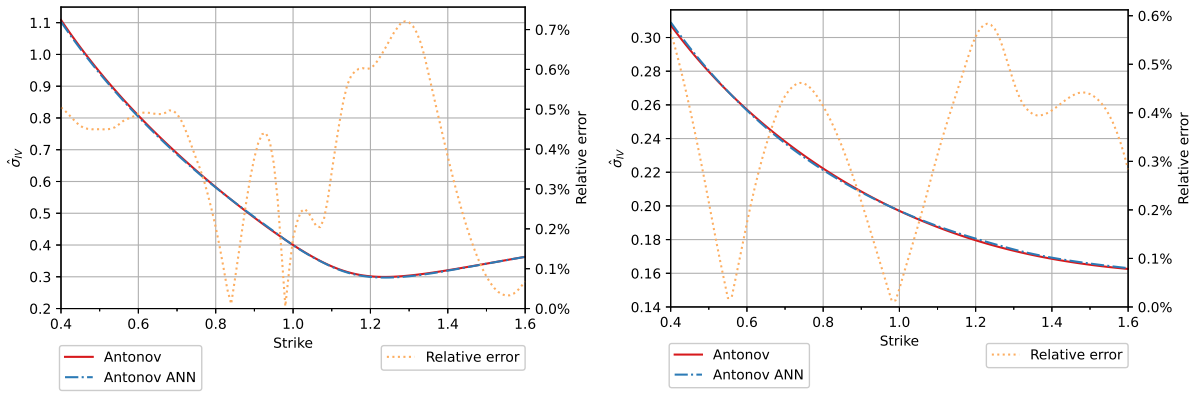
(b) Standard deviation of relative error

Figure 1: Accuracy of the neural network with respect to the Antonov et al. (2013) implied volatility. The relative error of the neural network approximation is calculated on five million randomly generated samples.

Interestingly, relatively high errors can be observed around at the money for short term options as well. The Antonov approximation is not defined at the exact at the money point

($K = 1$) as elaborated upon in Appendix A. This results in some unstable behaviour around at the money for short term options with a low initial volatility α. We also observe higher standard deviations for the relative errors around at the money for all maturities, with the standard deviation highest at the short-term options.

To illustrate the neural network accuracy in practice, we plot the Antonov et al. (2013) and neural network implied volatilities for a short term and long term IV smile in Figure 2 below. Both the short and long term smiles are approximated closely by the neural network, with a relative error of about 0.7% maximum.



(a) $T = 10$ days, $\alpha = 0.40$, $\beta = 0.50$, $\rho = -0.70$, $v = 2.00$ (b) $T = 9.5$ years, $\alpha = 0.20$, $\beta = 0.50$, $\rho = -0.50$, $v = 0.30$

Figure 2: Antonov et al. (2013) implied volatility smile with the neural network approximation (left axis) together with the relative error of the neural network approximation (right axis). A short term (a) and long term (b) implied volatility smile is considered.

5.2 Numerical experiment

We compare the accuracy and speed of the neural network IV approximation using simulated data. We consider the 10,000 randomly sampled SABR parameter sets as given. We simulate 10,000 implied volatility smiles as well as the SABR greeks for strikes between 0.40 and 1.60. For each strike, we calculate the relative error of the IV approximation compared with the Monte Carlo simulated implied volatility. We perform the same analysis for the SABR greeks.

Figure 3 shows the average relative errors of the approximated implied volatility. The Hagan approximation performs poorly compared to the other methods, with the average relative error not below 13%. The error increases for lower strikes (more in the money options). In contrast, the relative errors of the Antonov et al. (2013) and neural network approximations are much lower and nearly identical at around 4%. The error of the latter approximations depends less on the strike price, increasing only slightly for more in the money options.

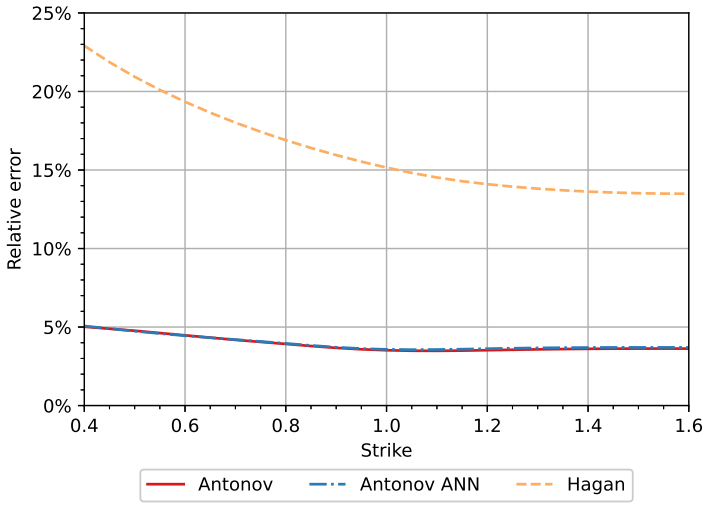


Figure 3: Average relative error of the SABR implied volatility with respect to the Monte Carlo implied volatility. The average is taken using 10,000 random SABR parameter sets. The relative errors are calculated for the Antonov et al. (2013) approximation, neural network approximation and Hagan et al. (2002) approximation.

The relative errors of the estimated greeks with respect to the Monte Carlo greeks are shown in Figure 16 in Appendix C. For all methods, the relative error of the delta hedge increases with the strike price. The Antonov relative delta errors are lowest, starting at around 1% for the lower strikes and rising to about 4%. The neural network errors start only slightly higher than the Antonov errors but increase sharply around the 1.3 strike to the point that the errors are higher than the Hagan errors at the 1.5 strike. Before the 1.5 strike, the Hagan errors are highest, starting with a relative error of 1.5% and moving to 5% for the 1.6 strike.

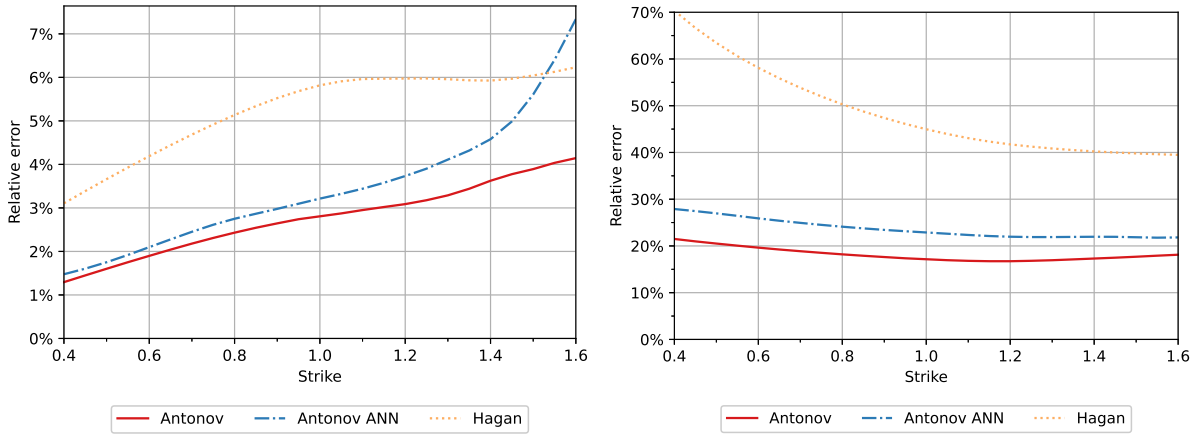
For the vega hedge, the neural network and Antonov et al. (2013) approximations have the lowest relative errors. Although the errors are close to each other, the neural network actually performs slightly better for the 0.8 strike upwards. The Hagan errors are about 10 percentage points higher over the complete strike range.

Table 1: Average time to approximate an implied volatility smile for the 25 strikes $0.40, 0.45, ..., 1.55, 1.60$. The average is taken using 10,000 random SABR parameter sets for the Antonov et al. (2013) approximation, neural network approximation and Hagan et al. (2002) approximation.

	Antonov	Antonov ANN	Hagan
time (s)	0.77	$4.44 \times 10^{-}3$	$2.13 \times 10^{-}3$

Table 1 shows the average time needed to calculate the implied volatility smile for the 25 strikes. The speed of the neural network implied volatility approximation now becomes apparent, performing over 120 times faster than the Antonov approximation. The network is about twice as slow compared with the Hagan implied volatility approximation.

We next consider the 10,000 parameter sets as unknown and calibrate the simulated smiles using the neural network, Antonov and Hagan approximations. With the calibrated SABR parameters, we calculate the delta and vega hedges.



(a) Delta

(b) Vega

Figure 4: Average relative error of the SABR greeks with respect to the Monte Carlo greeks. The average is taken using 10,000 random SABR parameter sets. The greeks are calculated using the Antonov et al. (2013) approximation, Antonov neural network approximation and Hagan et al. (2002) approximation.

Figure 4 shows the average relative errors of the calibrated delta and vega hedges with respect to the Monte Carlo hedges. When calibrated to simulated smiles, the neural network and Antonov et al. (2013) approximations best manage to approximate the delta hedge. The neural network error increases more for higher strikes, such that for strikes above 1.5 the error is larger than the Hagan relative error. For lower strikes, the neural network and Antonov errors are considerably lower. This pattern is similar to the previous experiment when we considered the parameters to be known, see Figure 16 (a) in Appendix C.

The Hagan approximation performs poor for the vega hedge as well. The average relative error starts at around 70% for the lower strikes and decreases to approximately 40% for the higher strikes. The relative vega error for Antonov is below 20% for most of the strikes, and the neural network relative error between 20% and 30%, both well below the Hagan errors.

Table 2: Average time to calibrate an implied volatility smile for the 25 strikes $0.40, 0.45, ..., 1.55, 1.60$. The average is taken using 10,000 random SABR parameter sets for the Antonov et al. (2013) approximation, neural network approximation and Hagan et al. (2002) approximation.

	Antonov	Antonov ANN	Hagan
time (s)	75.36	0.28	0.08

Table 2 displays the average calibration times over all simulated smiles. We observe a considerable decrease in calibration time for the neural network compared to the Antonov approx-

imation. The neural network calibration is over 250 times faster than the Antonov calibration, the latter on average taking over a minute to complete. With 0.28 seconds calibration time, the network is about 3.5 times slower compared with the Hagan et al. (2002) approximation, but both fast regardless.

The numerical experiment shows the power of the neural network approach. The network achieves similar accuracy as the Antonov approximation in implied volatility, delta, and vega hedges when the SABR parameters are considered to be known. Even when the parameters need to be calibrated, the network performs much better than the traditional Hagan approximations while the calibration is still fast. The neural network can calibrate to an implied volatility smile more accurately than the Hagan approximation based on the delta and vega hedges. We observe a slight decrease in accuracy with respect to the Antonov approximation, but speed increases of 100 to 250 times for the neural network. The only exception to the accuracy improvements is the neural network delta approximation for high strikes, performing worse than the Hagan approximations for strikes above 1.5.

5.3 S&P500 calibration

To examine the practical applicability of the neural network, we calibrate all methods on S&P500 option data. We start with the S&P500 implied volatility surface on 5 August 2019. We consider options maturing in 7 days or longer, which totals to 33 expiries. To make calibration with Antonov et al. (2013) feasible, we take a total of 25 strikes for each expiration. The calibrated SABR parameters for each method can be found in Figure 17 in Appendix C. The calibrated values for α and v are close for the Hagan et al. (2002) and neural network approximations. The Antonov et al. (2013) approximation shows some deviations in calibrated α and v for shorter maturities. For correlation ρ, not much consistency is found. Especially the correlation parameters calibrated by the neural network deviate a lot across the maturities.

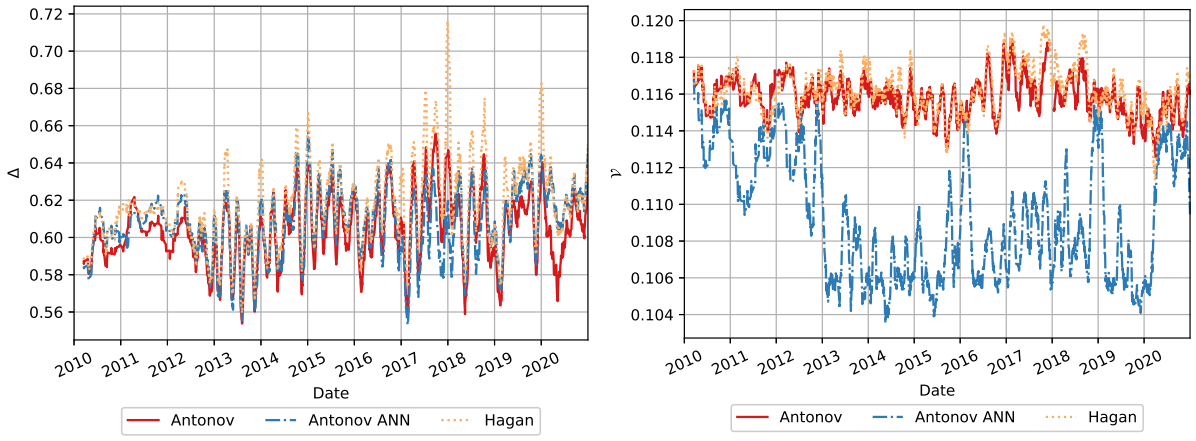
Table 3: Calibration time for the S&P500 implied volatility surface on 5 August 2019 for the Antonov et al. (2013) approximation, neural network approximation and Hagan et al. (2002) approximation.

	Antonov	Antonov ANN	Hagan
average time per smile (s)	25.58	0.18	0.07
total calibration time (s)	818.44	6.02	2.41

We show the average calibration time over all expiries in Table 3. Again, the neural network calibration is over 140 times faster compared to the Antonov calibration. The Hagan approximation is fastest; calibration to the entire surface takes around 2.4 seconds, whereas a full surface calibration by Antonov takes over 13 minutes. The neural network calibration is

about 2.5 times slower than the Hagan calibration, with about six seconds for the entire surface.

Next, we consider daily S&P500 options maturing in 30 days between 1 January 2010 and 31 December 2020.[3] The 20-day moving averages of the calibrated parameters are plotted in Figure 18 in Appendix C. The Hagan and Antonov approximations find approximately the same values for all parameters over the whole sample. The neural network calibrates the same values for initial volatility α as well, but we do see differences in the calibrated correlation ρ and volatility of volatility v. In general, ρ is calibrated lower, and v is calibrated higher by the neural network.



(a) Delta (Δ)

(b) Vega ($\mathcal{V}$)

Figure 5: 20-day moving averages of at the money SABR delta and vega hedges calibrated on S&P500 options maturing in 30 days. Calibration has been performed using the Antonov et al. (2013) approximation, neural network approximation and the Hagan et al. (2002) approximation.

With the calibrated parameters, we calculate the at the money delta and vega hedges. Figure 5 shows the 20-day moving averages of the calibrated greeks. We don't observe large differences between the methods for the delta hedges, although the Hagan estimates have higher upward spikes. However, we do see significant discrepancies in the calibrated vega hedges. The neural network vegas are considerably lower over the entire sample period, whereas the Antonov and Hagan vega hedges are very similar. This is surprising because a similar performance to the Antonov approximation is expected for the neural network. Furthermore, the Antonov and Hagan vega approximations are less volatile compared with the neural network approximations.

The results from the S&P500 calibration experiment again show the speed of the neural network approximation. However, comparing the calibrated delta and vega hedges shows indications of unstable behaviour of the neural network on market data. The differences seem to be explained

[3] For each trading day, we consider all options maturing between 25 and 35 days and choose the most liquid maturity.

because the neural network in general calibrates correlation ρ to lower values and volatility of volatility v to higher values, affecting the estimated vega hedges. The differences are surprising since the numerical experiment suggested similar calibration performance for the neural network and Antonov approximation.

5.4 Bayesian calibration

We start the Bayesian calibration experiment with a numerical example. We simulate an implied volatility smile for $T = 30$ days, $\alpha = 0.50$, $\rho = -0.60$, $v = 3.0$. The smile, together with the neural network and Hagan et al. (2002) approximations, is shown in Figure 6.

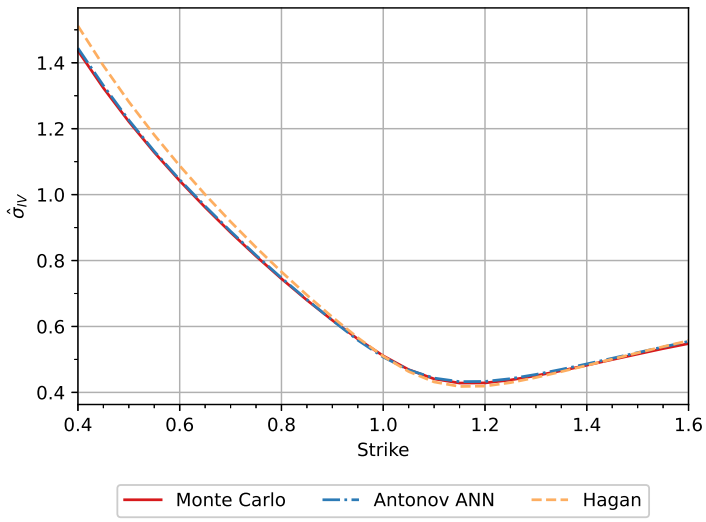
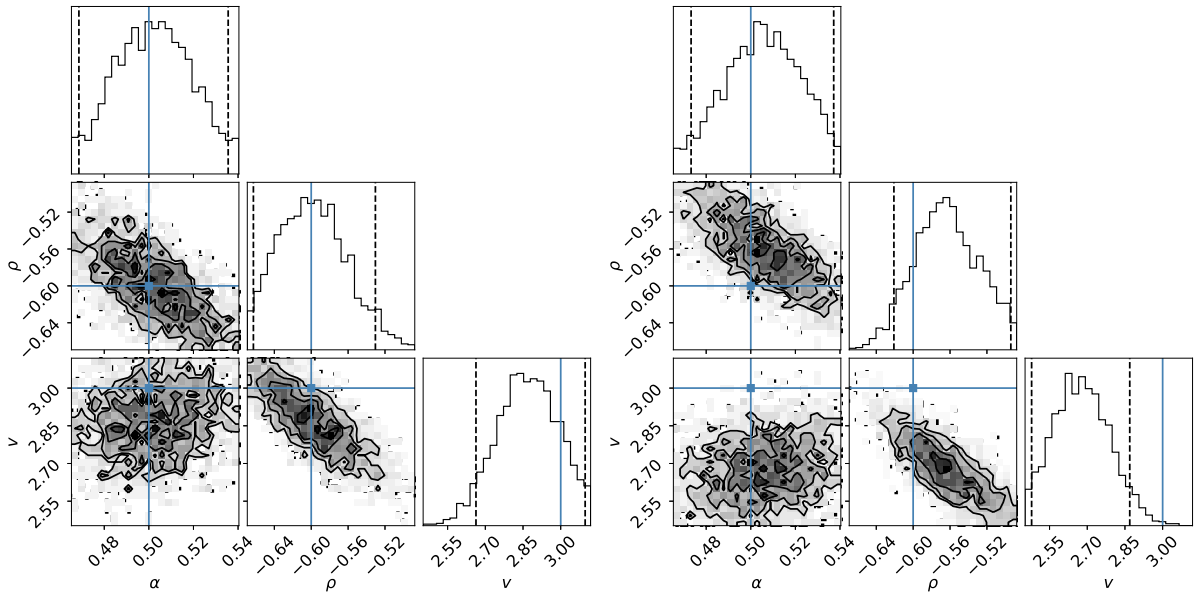


Figure 6: Monte Carlo implied volatility smile corresponding to $T = 30$ days, $\alpha = 0.50$, $\rho = -0.60$, $v = 3.0$ with the Antonov et al. (2013) neural network approximation and Hagan et al. (2002) approximation.

The Hagan et al. (2002) approximation is unable to closely represent the Monte Carlo implied volatilities for the lower strikes, thus we would expect errors in the calibrated parameters. We perform the nonlinear Bayesian regression using the Monte Carlo variance for the variance of the error terms in Equation (9). In Figure 7, the histograms of the marginal posterior distribution of the SABR parameters are plotted together with the contour plots of the two-dimensional marginal posterior distributions of all pairs of SABR parameters.[4] In general, the marginal posteriors of the neural network fall better around the real parameter values than the posteriors of the Hagan et al. (2002) approximation. The Hagan marginal posterior slightly overestimates ρ, whereas the neural network posterior is much less biased. For volatility of volatility v, the neural network posterior is quite accurate. This contrasts with the Hagan posterior, where the real parameter value does not fall in the 5% and 95% quantiles (denoted with dashed vertical

[4] All plots have been created using Foreman-Mackey (2016).

lines). The marginal posterior distribution of the initial volatility α does fall around the real value for both methods.

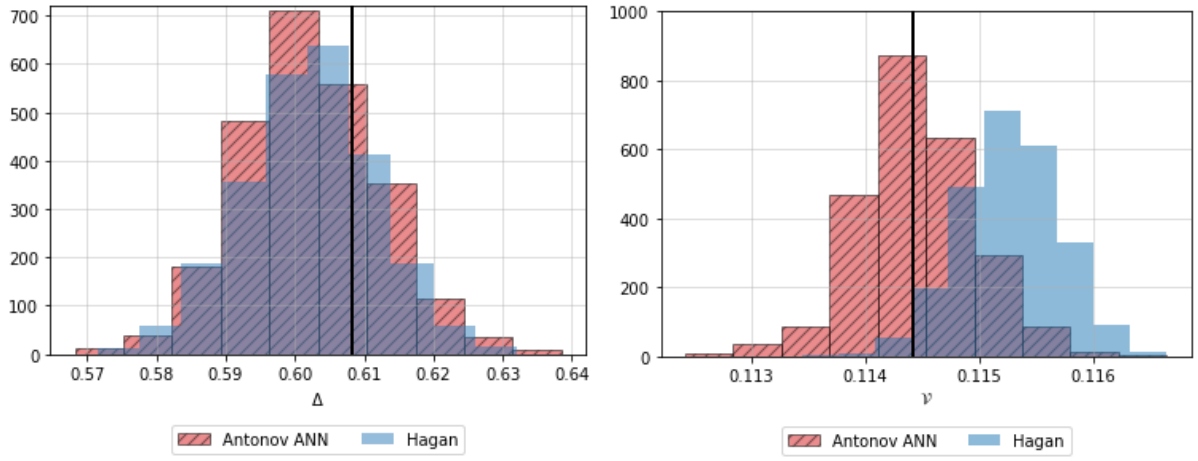


(a) Neural network approximation

(b) Hagan et al. (2002) approximation

Figure 7: Marginal posterior distributions of the SABR parameters in a nonlinear Bayesian regression on a Monte Carlo simulated implied volatility smile using $T = 30$ days, $\alpha = 0.50$, $\rho = -0.60$, $v = 3.0$. The regression has been performed using Antonov et al. (2013) neural network approximated implied volatilities (a) and Hagan et al. (2002) implied volatilities (b). The dashed vertical lines indicate the 5% and 95% quantiles. The solid blue vertical line indicates the real parameter value.

For all parameter pairs simulated from the posterior distribution, we calculate the SABR delta and vega hedges using Equations (7) and (8). These can be found in the histograms in Figure 8 below.



(a) SABR delta (Δ)

(b) SABR vega ($\mathcal{V}$)

Figure 8: At the money delta and vega hedges corresponding to parameters simulated from the posterior distribution of a nonlinear Bayesian regression on a Monte Carlo simulated implied volatility smile. The smile is simulated using $T = 30$ days, $\alpha = 0.30$, $\rho = -0.60$, $v = 3.0$. The vertical lines indicate the true delta and vega values.

Both the neural network and the Hagan approximation are able to estimate the delta hedge correctly. This is somewhat surprising since the Hagan posterior parameter pairs are sampled worse compared to the neural network. We see larger differences for the vega hedge, which the neural network estimates more accurately. For both the marginal posterior sampled parameters and the corresponding greeks, we do not observe notable differences in the variance of the samples. Therefore, the neural network calibration is preferred for this simulated smile since the SABR parameters and SABR vega are estimated more accurately.

For completeness, we also consider an implied volatility smile which the Hagan et al. (2002) approximation can match more closely. We now take $T = 30$ days, $\alpha = 0.40$, $\rho = 0.30$, $v = 2.0$. The smile is plotted in Figure 19 in Appendix C. The posterior histograms and contour plots can be found in Figure 20 in Appendix C. Now, both methods perform nearly identical; all posterior distributions fall nicely around the real parameter values. Moreover, we do not observe large differences in the variance of the marginal posterior distributions such that a clear preference between the methods is hard to justify. The histograms of the SABR delta and vega values can be found Figure 21 in Appendix C. The posterior sampled parameters correctly estimate the delta hedge. The Hagan vega hedge approximation is slightly underestimated. However, the variance is lower compared with the neural network vega hedges.

We finally apply the Bayesian calibration experiment on S&P500 options on 4 May 2020. We consider 25 strikes of the options maturing on 1 June 2020, thus maturing in 28 days. Figure 9 shows the implied volatility of the options, with the calibrated SABR implied volatility for the neural network and Hagan approximations. Both methods fit the data nearly identical.

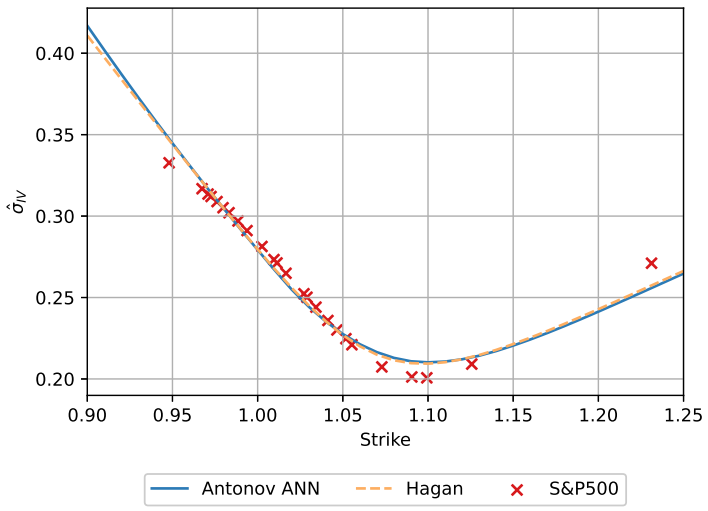
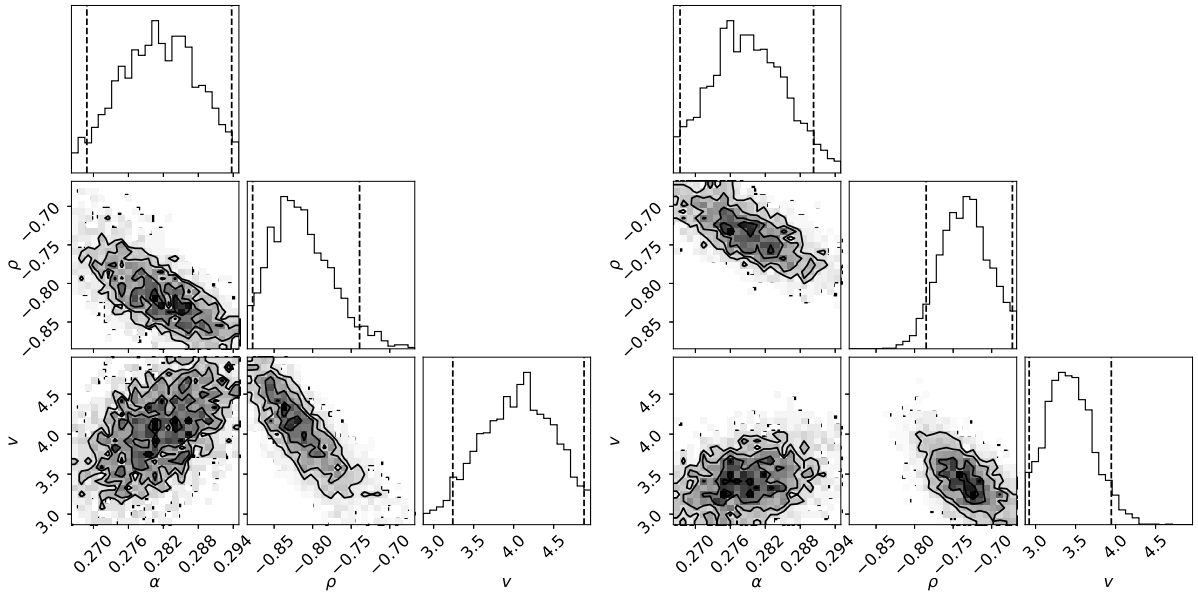


Figure 9: Implied volatilities of S&P500 options on 4 May 2020 maturing on 1 June 2020. The implied volatility smile for the neural network and Hagan et al. (2002) approximation is obtained using traditional calibration.



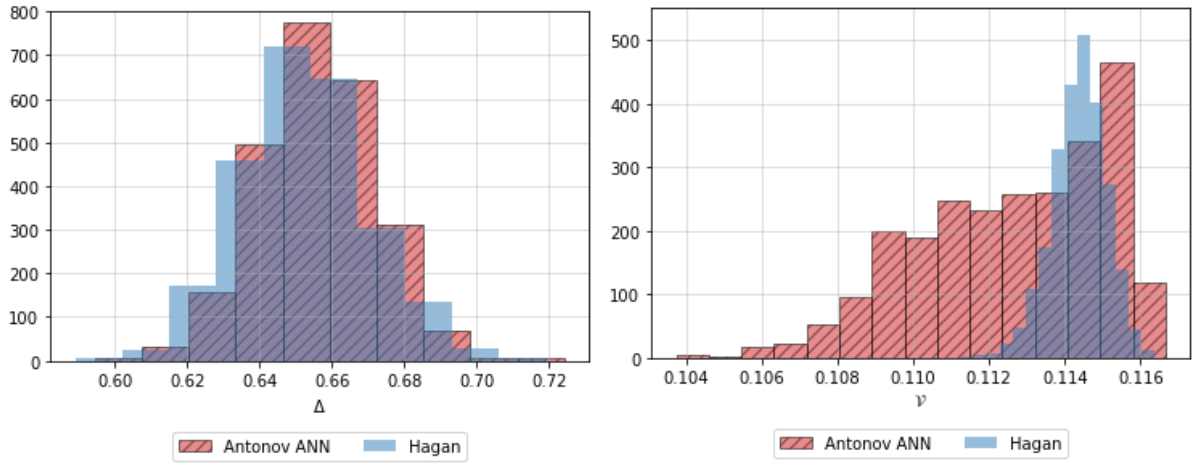
(a) Neural network approximation

(b) Hagan et al. (2002) approximation

Figure 10: Marginal posterior distributions of the SABR parameters in a nonlinear Bayesian regression on the implied volatility smile of S&P500 options on 4 May 2020. The expiry date of the options is 1 June 2020. The regression has been performed using Antonov et al. (2013) neural network approximated implied volatilities (a) and Hagan et al. (2002) implied volatilities (b). The dashed vertical lines indicate the 5% and 95% quantiles.

Figure 10 shows the posterior histograms and contour plots of the parameters. We observe a pattern similar to the S&P500 calibration experiment in Section 5.3; the posterior sampled correlation ρ is generally lower for the neural network. The posterior sampled volatility of volatility v is higher for the network compared with the Hagan approximation. More striking is the uncertainty in the parameters. The neural network sampled posterior v has a much higher variance than the Hagan sampled parameters. We also observe higher uncertainty in the sampled correlation by the neural network, although to a lesser extent.

To examine the consequence of this uncertainty, we calculate the at the money delta and vega hedges. Figure 11 shows the histograms of the delta and vega hedges corresponding with the posterior sampled parameters. We do not observe notable differences in the estimated delta hedges of the neural network and Hagan approximations. However, the uncertainty in the neural network posterior parameters is showing clearly in the estimated vega hedges. The variance of the posterior vega is much higher for the neural network compared with the Hagan approximation.



(a) SABR delta (Δ)

(b) SABR vega ($\mathcal{V}$)

Figure 11: At the money delta and vega hedges corresponding to parameters simulated from the posterior distribution of a nonlinear Bayesian regression on the implied volatility smile of S&P500 options on 4 May 2020. The expiry date of the options is 1 June 2020. The vertical lines indicate the true delta and vega values.

The Bayesian calibration experiment emphasises the good performance of the neural network on simulated data. We find that the posterior sampled parameters and greeks of the neural network are at least as accurate as those sampled using the Hagan approximation. However, interesting differences appear when performing the Bayesian calibration experiment on S&P500 data. The neural network is significantly less confident in the calibration of the volatility of volatility parameter v. The higher parameter uncertainty also results in a higher uncertainty in the corresponding vega hedges. This observation aligns with the unstable vega hedges found in the S&P500 calibration experiment from Section 5.3.

Although the neural network vega hedges are more accurate than the Hagan vega hedges in a simulated environment, they also seem less stable when applied to real market data. Since the Antonov vega estimates in Section 5.3 are stable and similar to the Hagan approximation, the vega instability appears to be a result of the neural network approximation. The reason why the network performance degrades on market data is not directly clear. Looking at Figure 9, the S&P500 implied volatility smile has strikes between 0.90 and 1.25 whereas the network has been trained on strikes between 0.40 and 1.60. Therefore, it could be that the neural network accuracy is not optimal because it is trained on a much wider range of strike prices.

6 Conclusion and discussion

This research applies machine learning techniques to overcome the speed and accuracy tradeoff in SABR calibration. We approximate the computationally heavy but highly accurate SABR implied volatility map from Antonov et al. (2013) using an artificial neural network. Since the network is trained on a wide range of parameters and maturities, it can be applied to options on different asset classes without retraining the model. Moreover, by only approximating the computationally heavy pricing map, traditional SABR theory and calibration methods can still be applied.

We provide evidence through numerical experiments that neural networks have the potential to increase SABR calibration speed and accuracy significantly. On simulated data, the neural network outperforms the traditional Hagan et al. (2002) formulas in implied volatility accuracy. The network can better capture volatility smile dynamics by estimating greeks more accurately. Moreover, the network's calibration speed is much higher than the Antonov approximation it is based on.

However, further research should be conducted on the application of the neural network to real market data. SABR calibration on S&P500 options indicates instabilities in the vega hedges estimated by the network. We confirm this with a Bayesian calibration experiment, where higher variances in the calibrated volatility of volatility parameters are visible. The higher parameter uncertainty results in more uncertainty in the vega hedges as well. The instability seems to result from the neural network approximation since it is not directly found in the Antonov approximation. We will therefore offer suggestions for improvements in the neural network accuracy.

First, since the network is trained on strikes between 0.40 and 1.60 it could be that the instability on S&P500 data is imposed due to inaccuracies in the strike parameter K. In general, there are no liquid options on the S&P500 for extreme strikes as 0.40 or 1.60 (40% or 160% of the index value). By training a network on a smaller range of strike prices a higher neural network accuracy might be reached.

Second, it might also be beneficial to decrease the range of volatility of volatility parameter v. Although the range of v has bounds varying with time to maturity, they are set to be very wide in general. Decreasing the bounds of v could increase the network's accuracy for shorter maturities. Moreover, very high values for v cause unstable behaviour of the Antonov approximation, which could also influence the neural network training negatively.

Finally, since the neural network approach in this research is not limited to the Antonov et al. (2013) approximation, different implied volatility approximations or Monte Carlo simulations can be considered.

Looking forward, it is of interest whether the neural network approach might work better on other data, such as FX options or interest rate swaptions. Moreover, it could also be worth considering more elaborate evaluations of the methods. In this research, the standard SABR delta and vega hedges have been considered, however, the delta and vega hedges from Bartlett (2006) are more robust in theory. Finally, calibration performance on other greeks such as gamma, theta, vanna and volga is of interest.

References

Antonov, A., Konikov, M., and Spector, M. (2013). SABR spreads its wings. *Risk*, 26:58–63.

Antonov, A., Konikov, M., and Spector, M. (2015). The free boundary SABR: Natural extension to negative rates. *SSRN Electronic Journal*.

Bartlett, B. (2006). Hedging under SABR model. *Wilmott Magazine*, July/August:2–4.

Bayer, C., Horvath, B., Muguruza, A., Stemper, B., and Tomas, M. (2019). On deep calibration of (rough) stochastic volatility models.

Black, F. (1976). The pricing of commodity contracts. *Journal of Financial Economics*, 3(1-2):167–179.

Black, F. and Scholes, M. (1973). The pricing of options and corporate liabilities. *Journal of Political Economy*, 81(3):637–654.

Chen, B., Oosterlee, C., and van der Weide, H. (2012). A low-bias simulation scheme for the SABR stochastic volatility model. *International Journal of Theoretical and Applied Finance (IJTAF)*, 15:1250016–1.

Clevert, D., Unterthiner, T., and Hochreiter, S. (2016). Fast and accurate deep network learning by exponential linear units (ELUs). In Bengio, Y. and LeCun, Y., editors, *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

Cox, J. C. (1996). The constant elasticity of variance option pricing model. *The Journal of Portfolio Management*, 23(5):15–17.

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2:303–314.

Duembgen, M. and Rogers, L. C. G. (2014). Estimate nothing.

Dupire, B. (1994). Pricing with a smile. *Risk Magazine*, pages 18–20.

Eldan, R. and Shamir, O. (2016). The power of depth for feedforward neural networks.

Fernández, J., Ferreiro, A., García-Rodríguez, J., Leitao, A., López-Salas, J., and Vázquez, C. (2013). Static and dynamic SABR stochastic volatility models: Calibration and option pricing using GPUs. *Mathematics and Computers in Simulation*, 94:55–75.

Foreman-Mackey, D. (2016). corner.py: Scatterplot matrices in python. *The Journal of Open Source Software*, 1(2):24.

Foreman-Mackey, D., Hogg, D. W., Lang, D., and Goodman, J. (2013). emcee: The MCMC hammer. *Publications of the Astronomical Society of the Pacific*, 125(925):306â€“312.

Glorot, X., Bordes, A., and Bengio, Y. (2011). Deep sparse rectifier neural networks. In Gordon, G., Dunson, D., and Dudík, M., editors, *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pages 315–323, Fort Lauderdale, FL, USA. JMLR Workshop and Conference Proceedings.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press. `http://www.deeplearningbook.org`.

Hagan, P., Kumar, D., Lesniewski, A., and Woodward, D. (2002). Managing smile risk. *Wilmott Magazine*, 1:84–108.

Hagan, P. and Lesniewski, A. (2017). Bartlett's delta in the SABR model. *SSRN Electronic Journal*.

Hastie, T., Tibshirani, R., and Friedman, J. (2001). *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA.

Hernández, A. (2016). Model calibration with neural networks. *Neuroeconomics eJournal*.

Horvath, B., Muguruza, A., and Tomas, M. (2021). Deep learning volatility: a deep neural network perspective on pricing and calibration in (rough) volatility models. *Quantitative Finance*, 21(1):11–27.

Islah, O. (2009). Solving SABR in exact form and unifying it with LIBOR market model. *Derivatives*.

Jäckel, P. (2015). Let's be rational. *Wilmott*, 2014.

Keskar, N. S., Mudigere, D., Nocedal, J., Smelyanskiy, M., and Tang, P. T. P. (2017). On large-batch training for deep learning: Generalization gap and sharp minima.

Kingma, D. P. and Ba, J. (2017). Adam: A method for stochastic optimization.

Le Floc'h, F. and Kennedy, G. (2014). Explicit SABR calibration through simple expansions. *SSRN Electronic Journal*.

Leitao, Á., Grzelak, L. A., and Oosterlee, C. W. (2017a). On an efficient multiple time step Monte Carlo simulation of the SABR model. *Quantitative Finance*, 17(10):1549–1565.

Leitao, Á., Grzelak, L. A., and Oosterlee, K. (2017b). On a one time-step Monte Carlo simulation approach of the SABR model: Application to European options. *Applied Mathematics and Computation*, 293:461–479.

Leshno, M., Lin, V., Pinkus, A., and Schocken, S. (1993). Multilayer feedforward networks with a non-polynomial activation function can approximate any function. In *Neural Networks*, volume 6, pages 861–867.

Levenberg, K. (1944). A method for the solution of certain non-linear problems in least squares. *Quarterly of Applied Mathematics*, 2(2):164–168.

Lokvancic, M. (2020). Machine learning SABR model of stochastic volatility with lookup table. *SSRN Electronic Journal*.

Marquardt, D. W. (1963). An algorithm for least-squares estimation of nonlinear parameters. *Journal of the Society for Industrial and Applied Mathematics*, 11(2):431–441.

McGhee, W. A. (2011). An efficient implementation of stochastic volatility by the method of conditional integration. *ICBI Global Derivatives and Risk Management*.

McGhee, W. A. (2018). An artificial neural network representation of the SABR stochastic volatility model.

Obłój, J. (2008). Fine-tune your smile: Correction to Hagan et al.

Stoep, A., Grzelak, L., and Oosterlee, C. (2014). The time-dependent FX-SABR model: Efficient calibration based on effective parameters. *SSRN Electronic Journal*.

Zhang, M. and Fabozzi, F. J. (2016). On the estimation of the SABR model's beta parameter: The role of hedging in determining the beta parameter. *The Journal of Derivatives*, 24(1):48–57.

Appendix

A Antonov et al. (2013) call option approximation

In this section we describe the process of obtaining the Antonov et al. (2013) call option approximation $\hat{C}^{\text{Antonov}}$ for strike price K, forward price f_j, maturity T_j and SABR parameters α, β, ρ and v. First, we rewrite the SABR parameters to obtain a mimicking SABR model with zero correlation:

$$\begin{aligned} d\tilde{F}(t,T) &= \tilde{A}(t)\tilde{F}(t,T)^{\tilde{\beta}} d\tilde{W}_1(t) \qquad & \tilde{F}(0,T) &= f_j, \\ d\tilde{A}(t) &= \tilde{v}\tilde{A}(t) d\tilde{W}_2(t) & \tilde{A}(0) &= \tilde{\alpha} \\ d\tilde{W}_1(t) d\tilde{W}_2(t) &= 0. \end{aligned}$$

The parameters $\tilde{\alpha}, \tilde{\beta}$ and $\tilde{v}$ relate to the original SABR parameters α, β, ρ and v in the following way

$$\begin{aligned} \tilde{\beta} &= \beta \\ \tilde{v} &= \sqrt{v^2 + \frac{3}{2}\{v^2\rho^2 + \alpha v \rho(1-\beta) f_j^{\beta-1}\}} \\ \tilde{\alpha} &= \tilde{\alpha}^{(0)} + T_j \tilde{\alpha}^{(1)} \end{aligned} \tag{10}$$

where

$$\begin{aligned} \tilde{\alpha}^{(0)} &= \frac{2\Phi\delta\tilde{q}\tilde{v}}{\Phi^2 - 1}, \\ \delta\tilde{q} &= \frac{K^{1-\tilde{\beta}} - f_j^{1-\tilde{\beta}}}{1-\tilde{\beta}}, \\ \Phi &= \left(\frac{\alpha^{min} + \rho\alpha + v\delta q}{(1+\rho)\alpha} \right)^{\tilde{v}/v}, \end{aligned}$$

with

$$\begin{aligned} \alpha^{min} &= \sqrt{v^2\delta q^2 + 2\rho v \delta q \alpha + \alpha^2}, \\ q &= \frac{K^{1-\beta}}{1-\beta}, \\ \delta q &= \frac{K^{1-\beta} - f_j^{1-\beta}}{1-\beta}. \end{aligned} \tag{11}$$

$\tilde{\alpha}^{(1)}$ from Equation (10) is given by[5]

$$\tilde{\alpha}^{(1)} = \tilde{\alpha}^{(0)}\tilde{v}^2\sqrt{1+\tilde{R}^2}\frac{\frac{1}{2}\ln\left(\frac{\alpha\alpha^{min}}{\tilde{\alpha}^{(0)}\tilde{\alpha}^{min}}\right) - \mathcal{B}_{min}}{\tilde{R}\ln\left(\sqrt{1+\tilde{R}^2}+\tilde{R}\right)} \quad \text{for} \quad \tilde{R} = \frac{\delta q\tilde{v}}{\tilde{\alpha}^{(0)}}, \tag{12}$$

where

$$\begin{aligned}
\tilde{\alpha}^{min} &= \sqrt{\tilde{v}^2\delta q^2 + \left(\tilde{\alpha}^{(0)}\right)^2}, \\
\mathcal{B}_{min} &= -\frac{1}{2}\frac{\beta}{1-\beta}\frac{\rho}{\sqrt{1-\rho^2}}(\pi - \varphi - \arccos\rho - I), \\
\varphi &= \arccos\Big(-\frac{\delta qv + \alpha\rho}{\alpha^{min}}\Big).
\end{aligned}$$

Here,

$$I = \begin{cases} \frac{2}{\sqrt{1-L^2}}\left(\arctan\left(\frac{\mu_0+L}{\sqrt{1-L^2}}\right) - \arctan\left(\frac{L}{\sqrt{1-L^2}}\right)\right) & \text{for } L < 1 \\ \frac{1}{\sqrt{1-L^2}}\ln\frac{\mu_0(L+\sqrt{L^2-1})+1}{\mu_0(L-\sqrt{L^2-1})+1} & \text{for } L > 1 \end{cases},$$

with

$$\mu_0 = \frac{\delta qv\rho + \alpha - \alpha^{min}}{\delta qv\sqrt{1-\rho^2}}, \quad L = \frac{\alpha^{min}(1-\beta)}{K^{1-\beta}v\sqrt{1-\rho^2}}.$$

The forward price of the call option is now

$$\begin{aligned}
C^{\text{Antonov}}(K, f_j, T_j, \{\tilde{\alpha}, \tilde{\beta}, \tilde{v}\}) = (f_j - K)^+ + \frac{2}{\pi}\sqrt{f_j K}\Bigg\{ &\int_{s_-}^{s_+} \frac{\sin(\eta\phi(s))}{\sinh s} G(T_j\tilde{v}^2, s)ds \\
&+ \sin(\eta\pi)\int_{s_+}^{\infty} \frac{e^{-\eta\psi(s)}}{\sinh s} G(T_j\tilde{v}^2, s)ds\Bigg\},
\end{aligned} \tag{13}$$

where the integration limits are defined as

$$s_- = \operatorname{arcsinh}\left(\frac{\tilde{v}|q-q_0|}{\tilde{\alpha}}\right), \quad s_+ = \operatorname{arcsinh}\left(\frac{\tilde{v}|q+q_0|}{\tilde{\alpha}}\right),$$

with q from (11) and

$$q_0 = \frac{f_j^{1-\beta}}{1-\beta}.$$

[5] We note that Equation (12) is not equal to the equation for $\tilde{\alpha}^{(1)}$ from Antonov et al. (2013). The original equation from Antonov et al. (2013) does not give satisfactory results. In fact, by using the original equation it is not possible to replicate the numerical experiment in Antonov et al. (2013). Instead, Equation (12) is found in Antonov et al. (2015) and, surprisingly, is able to replicate the results from Antonov et al. (2013). Why the two formulas differ, and why a seemingly incorrect version is given in Antonov et al. (2013) is not clear and beyond the scope of this research.

Furthermore, for Equation (13) we need

$$\phi(s) = 2\arctan\sqrt{\frac{\sinh^2(s) - \sinh^2(s_-)}{\sinh^2(s_+) - \sinh^2(s)}},$$
$$\psi(s) = 2\text{arctanh}\sqrt{\frac{\sinh^2(s) - \sinh^2(s_+)}{\sinh^2(s) - \sinh^2(s_-)}},$$

and

$$\eta = \left|\frac{1}{2(\tilde{\beta})}\right|.$$

Finally, the kernel function $G(\tau, s)$ is given by

$$G(\tau, s) = 2\sqrt{2}\frac{e^{-\frac{\tau}{8}}}{\tau\sqrt{2\pi\tau}}\int_s^\infty ue^{-\frac{u^2}{\tau}}\sqrt{\cosh(u) - \cosh(s)}du.$$

An approximation of the kernel function is given in Antonov et al. (2013) such that Equation (13) only consists of single integrals:

$$G(\tau, s) \simeq \sqrt{\frac{\sinh(s)}{s}}e^{-\frac{s^2}{2\tau}-\frac{\tau}{8}}(R(\tau, s) + \delta R(\tau, s)),$$

where

$$\begin{aligned} R(\tau, s) &= 1 + \frac{3\tau g(s)}{8s^2} - \frac{5\tau^2(-8s^2 + 3g^2(s) + 24g(s))}{128s^4} \\ &\quad + \frac{35\tau^3\big(40s^2 + g^3(s) + 24g^2(s) + 120g(s)\big)}{1024s^6}, \\ g(s) &= s\coth(s) - 1, \\ \delta R(\tau, s) &= e^{\frac{\tau}{8}} - \frac{3072 + 384\tau + 24\tau^2 + \tau^3}{3072}. \end{aligned}$$

In this research, we only make use of the approximated kernel function as the numerical experiment in Antonov et al. (2013) shows that the approximation comes at the cost of at most 0.3 basispoints. After the parameters $\tilde{\alpha}, \tilde{\beta}$ and $\tilde{v}$ have been obtained from the original SABR parameters α, β, ρ and v the call option price can be approximated by numerical integration of Equation (13).

We note that the approximation is not defined for $K \approx f_j$. We solve this using linear interpolation between two points around at the money.

B Calculation of $\frac{\partial \hat{\sigma}_{IV}}{\partial f}$ with $\hat{\sigma}_{IV}^{\mathbf{ANN}}$

In this section we describe the process of approximating $\frac{\partial \hat{\sigma}_{IV}}{\partial f}$ used in Equation (7) with central differences. We write $\hat{\sigma}_{IV}(\zeta, \Theta)$ with $\zeta = \{K, T, f\}$ and $\Theta = \{\alpha, \rho, v\}$ as $\hat{\sigma}_{IV}\big(\{K, T, f\}, \{\alpha, \rho, v\}\big)$. To approximate its partial derivative with respect to f we usually compute

$$\frac{\hat{\sigma}_{IV}\big(\{K, T, f+\varepsilon\}, \{\alpha, \rho, v\}\big) - \hat{\sigma}_{IV}\big(\{K, T, f-\varepsilon\}, \{\alpha, \rho, v\}\big)}{2\varepsilon}$$

for sufficiently small ε. However, computing $\hat{\sigma}_{IV}\big(\{K, T, f+\varepsilon\}, \{\alpha, \rho, v\}\big)$ (and its analog) is not possible for the neural network approach since $\hat{\sigma}_{IV}^{\text{ANN}}$ is only defined with $f = 1$.

We describe the calculation for $f + \varepsilon$. Instead of a shift in the forward price f, we consider an equivalent shift in the strike price K. In the first situation, the strike price relative to the shifted forward price was $\frac{K}{f+\varepsilon}$. Therefore, in the new situation we consider the strike price $\frac{fK}{f+\varepsilon}$. This way, when we fix the forward price, the relative strike price still equals $\frac{K}{f+\varepsilon}$.

Although we consider the same relative strike and forward price as the normal situation, we still have

$$\hat{\sigma}_{IV}\big(\{K, T, f+\varepsilon\}, \{\alpha, \rho, v\}\big) \neq \hat{\sigma}_{IV}\left(\Big\{\frac{fK}{f+\varepsilon}, T, f\Big\}, \{\alpha, \rho, v\}\right)$$

because the initial forward price on the right hand side is lower than the forward price on the left hand side. Therefore, we are now working with a SABR model with different initial parameters. This can be corrected for by multiplying the initial volatility α with $\big(\frac{f}{f+\varepsilon}\big)^{\beta}$, following from the forward price process from Equation (1):

$$\begin{aligned} dF(0,T) &= A(0)F(0,T)^{\beta} dW_1(0) \\ &= \alpha f^{\beta} dW_1(0) \\ &= \alpha f^{\beta}(f+\varepsilon)^{-\beta}(f+\varepsilon)^{\beta} dW_1(0) \\ &= \alpha \left(\frac{f}{f+\varepsilon}\right)^{\beta} (f+\varepsilon)^{\beta} dW_1(0). \end{aligned}$$

Therefore, the following holds

$$\hat{\sigma}_{IV}\big(\{K, T, f+\varepsilon\}, \{\alpha, \rho, v\}\big) = \hat{\sigma}_{IV}\left(\Big\{\frac{fK}{f+\varepsilon}, T, f\Big\}, \Big\{\alpha\Big(\frac{f}{f+\varepsilon}\Big)^{\beta}, \rho, v\Big\}\right),$$

such that $\frac{\partial \hat{\sigma}_{IV}}{\partial f}$ can be computed for the neural network implied volatility approximation $\hat{\sigma}_{IV}^{\text{ANN}}$.

C Figures

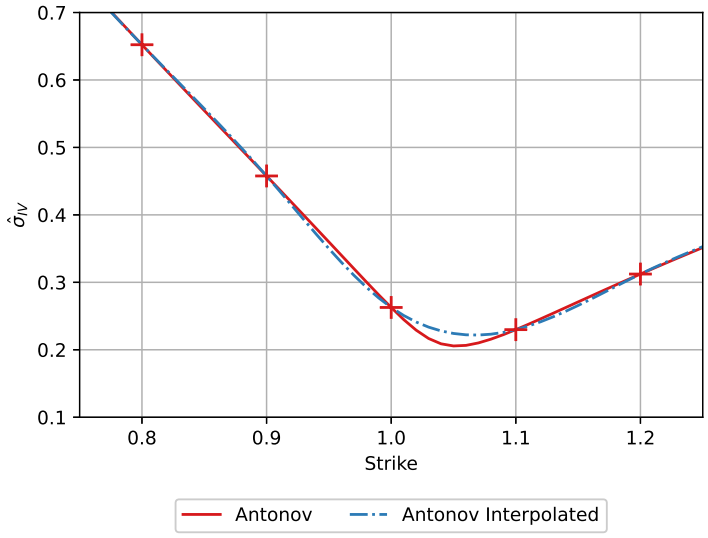


Figure 12: Demonstration of the inaccuracy imposed by interpolation when a heavy curve is present in the implied volatility smile. The solid line is an implied volatility smile constructed using Antonov et al. (2013) with SABR parameters $\alpha = 0.26$, $\beta = 0.50$, $\rho = -0.75$, $v = 5.40$ with 15 days to maturity. The dashed line is constructed using cubic spline interpolation on the strikes $0.8, 0.9, ..., 1.2$, creating inaccuracies around the curved part of the smile.

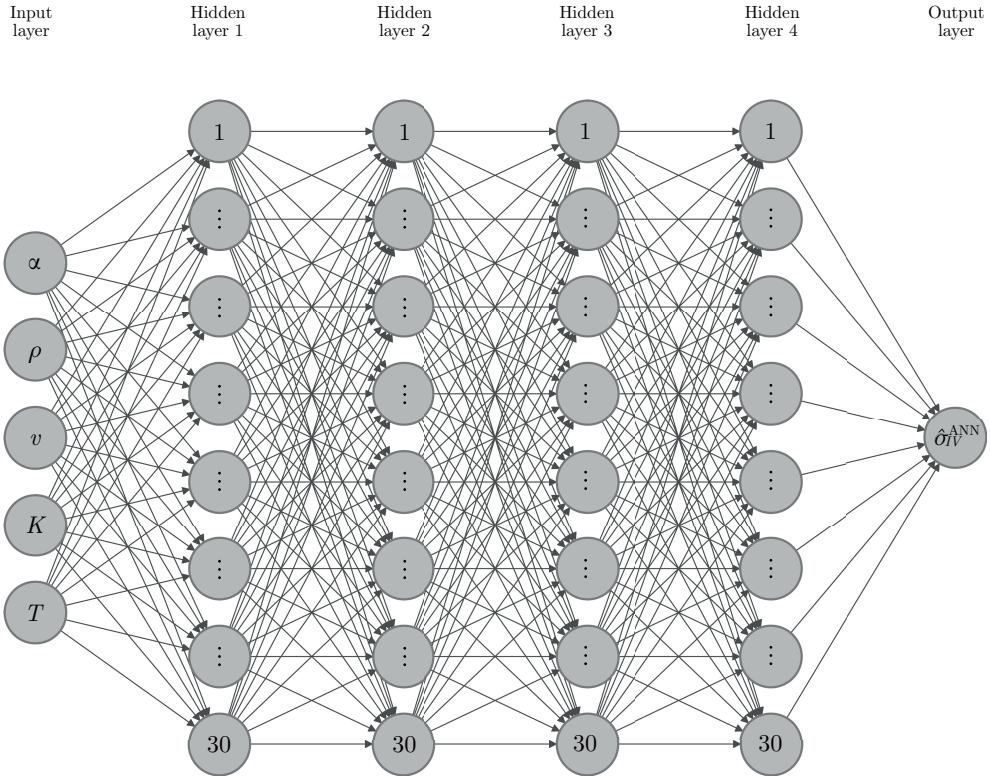
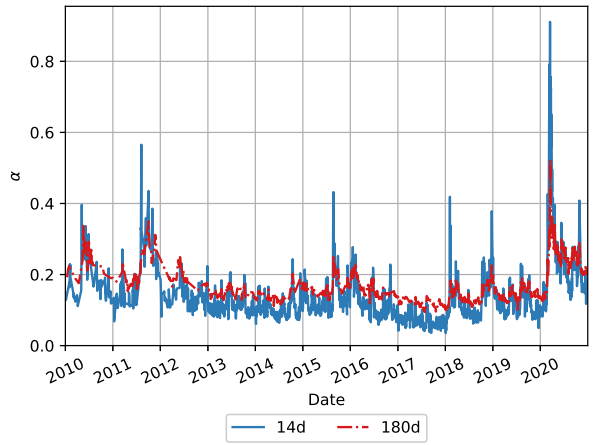
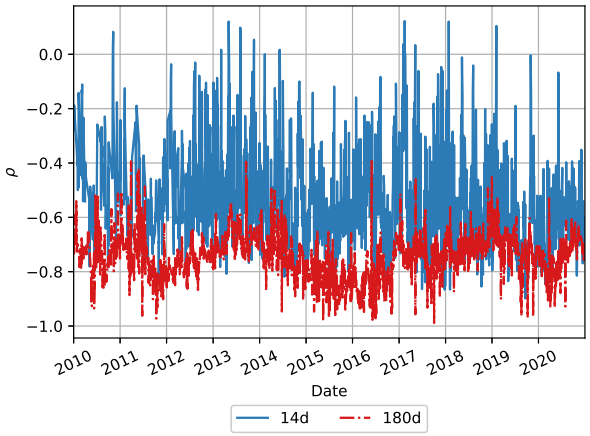


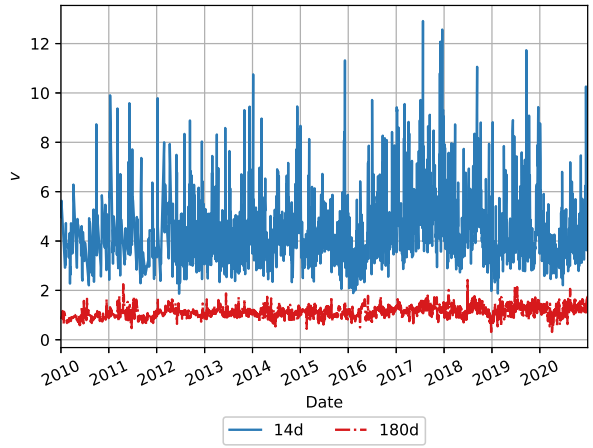
Figure 13: Schematic of the aritificial neural network architecture. We have 4 hidden layers all with 30 nodes, an input layer with 5 nodes and an output layer with a single node for the implied volatility $\hat{\sigma}_{IV}^{\text{ANN}}$.



(a) α



(b) ρ



(c) v

Figure 14: SABR parameters α, ρ, v calibrated using the Hagan et al. (2002) formula on S&P500 option data ranging from 1 January 2010 until 31 December 2020 with $\beta = 0.5$

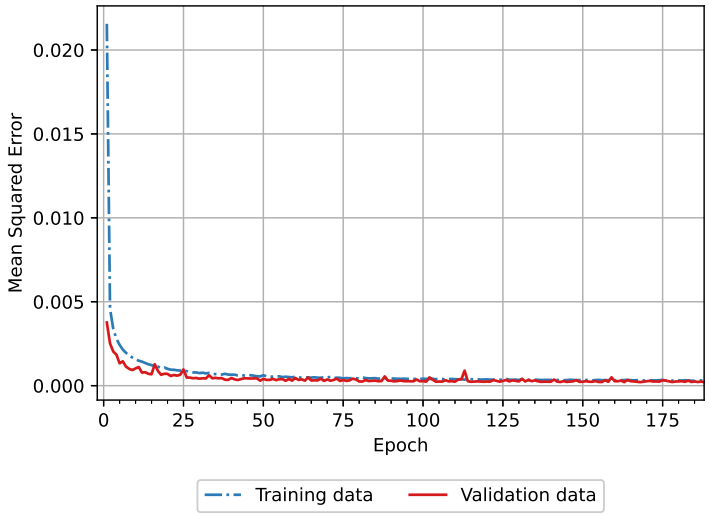
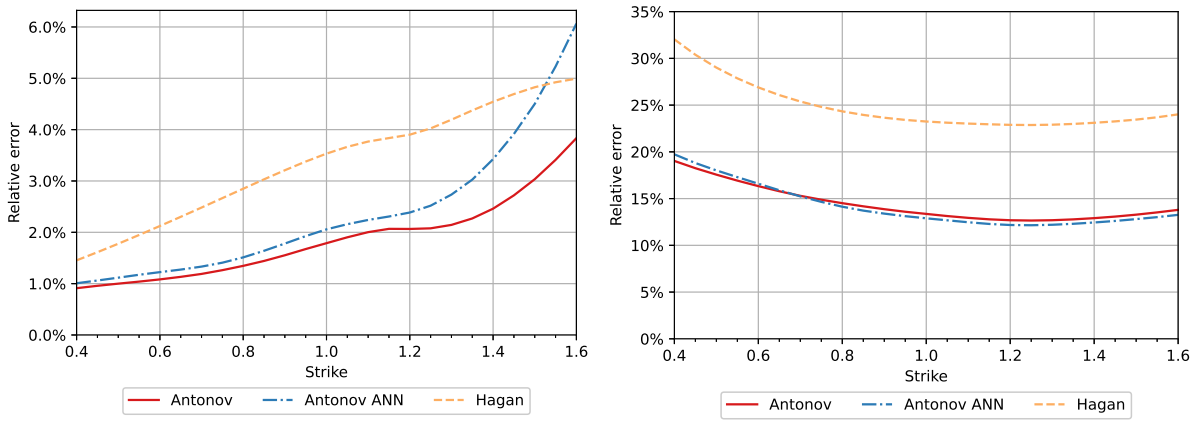


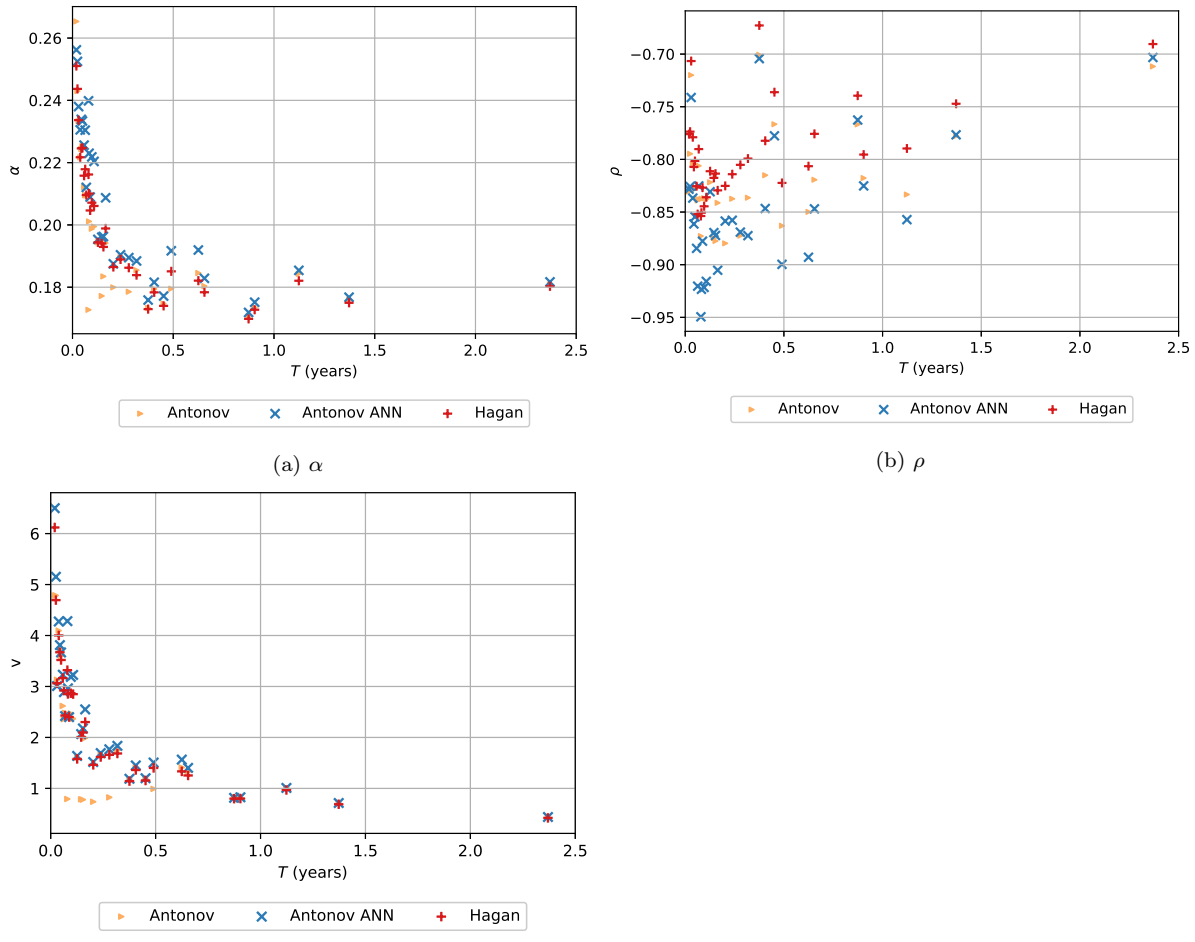
Figure 15: Mean squared error cost function of neural network training for the training data and validation data sets.



(a) Delta

(b) Vega

Figure 16: Average relative error of the SABR greeks with respect to the Monte Carlo greeks. The average is taken using 10,000 random SABR parameter sets. The greeks are calculated using the Antonov et al. (2013) approximation, the Antonov et al. (2013) neural network approximation and the Hagan et al. (2002) approximation.

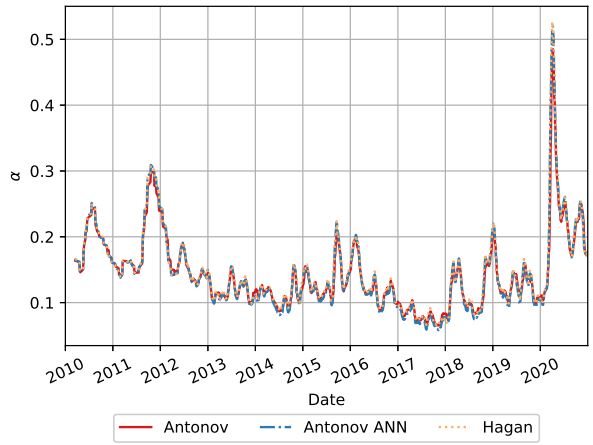


(a) α

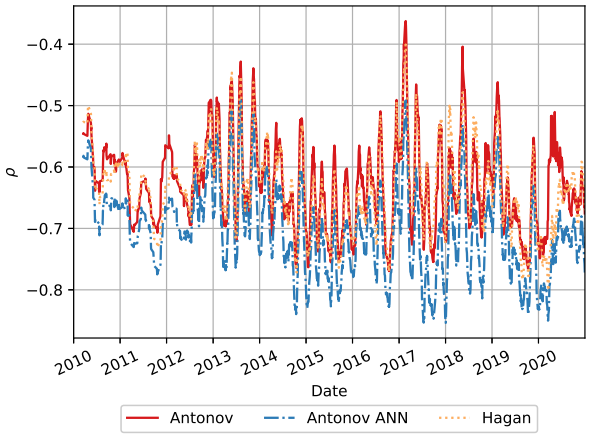
(b) ρ

(c) v

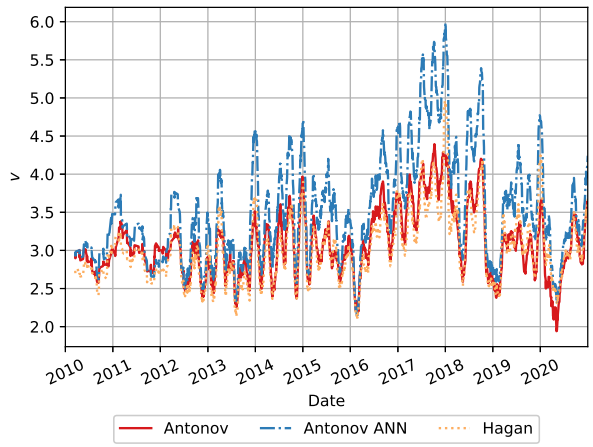
Figure 17: SABR parameters α, ρ, v calibrated using the Antonov et al. (2013) implied volatility approximation, Antonov et al. (2013) neural network approximation and Hagan et al. (2002) approximation. The model is calibrated on the S&P500 implied volatility surface on August 5th 2019.



(a) α



(b) ρ



(c) v

Figure 18: 20-day moving average of SABR parameters α, ρ, v calibrated on S&P500 options maturing in 30 days between 1 January 2010 and 31 December 2020. The calibration has been performed using the Antonov et al. (2013) approximation, neural network approximation and the Hagan et al. (2002) approximation.

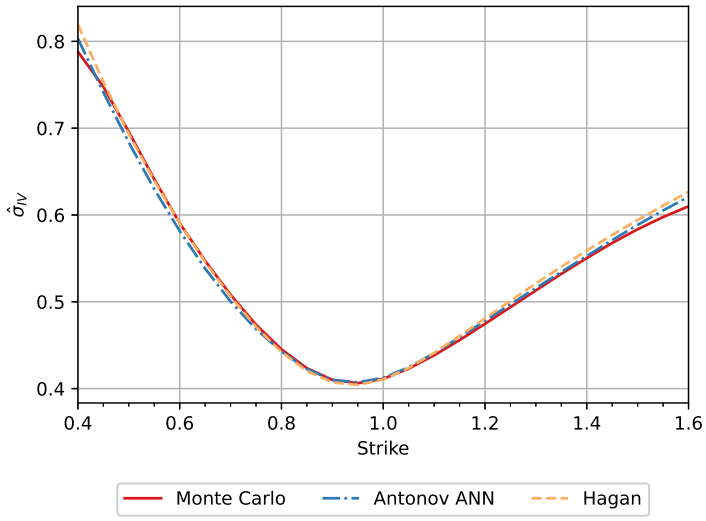
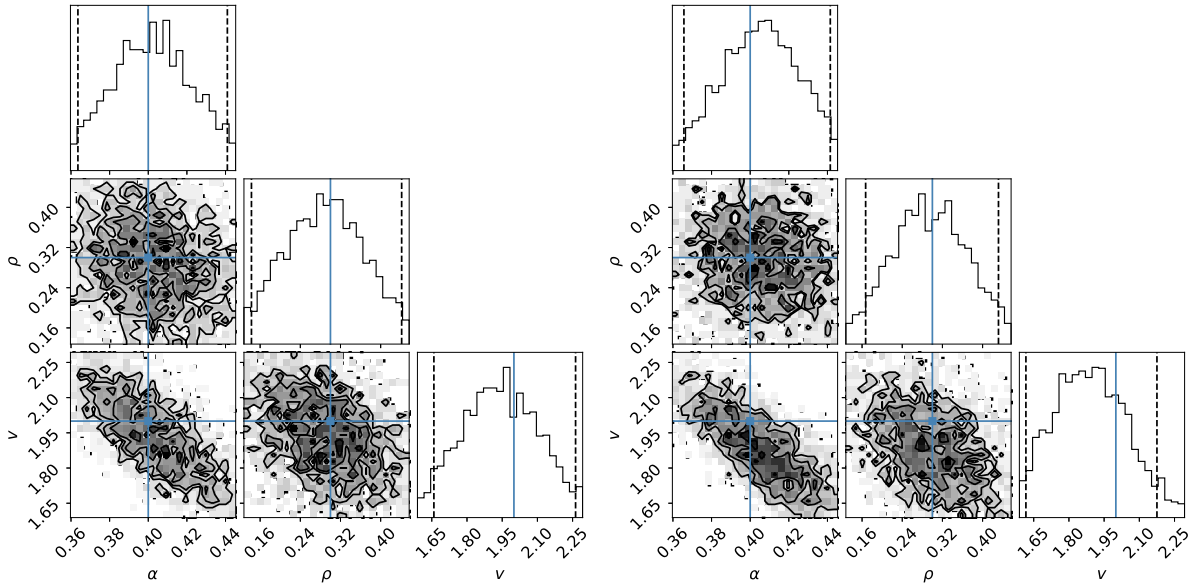


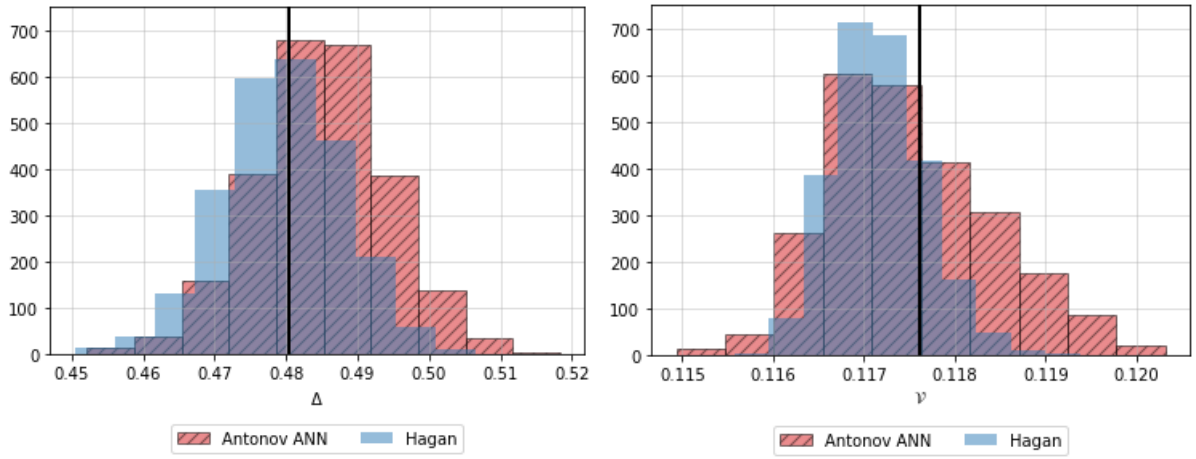
Figure 19: Monte Carlo implied volatility smile corresponding with $T = 30$ days, $\alpha = 0.40$, $\rho = 0.30$, $v = 2.0$ with the Antonov et al. (2013) neural network approximation and Hagan et al. (2002) approximation.



(a) Neural network approximation

(b) Hagan et al. (2002) approximation

Figure 20: Marginal posterior distributions of the SABR parameters in a nonlinear Bayesian regression on a Monte Carlo simulated implied volatility smile using $T = 30$ days, $\alpha = 0.40$, $\rho = 0.30$, $v = 2.0$. The regression has been performed using Antonov et al. (2013) neural network approximated implied volatilities (a) and Hagan et al. (2002) implied volatilities (b). The dashed vertical lines indicate the 5% and 95% quantiles, the solid blue vertical line indicates the real parameter value.



(a) SABR delta (Δ)

(b) SABR vega ($\mathcal{V}$)

Figure 21: At the money delta and vega hedges corresponding to parameters simulated from the posterior distribution of a nonlinear Bayesian regression on a Monte Carlo simulated implied volatility smile. The smile is simulated using $T = 30$ days, $\alpha = 0.40$, $\rho = 0.30$, $v = 2.0$. The vertical lines indicate the true delta and vega values.