Implied volatility surfaces in time, learned by a neural network

For traders and market makers in financial markets, the implied volatility surface is a very useful tool. Traders, for example, use implied volatilities in option pricing to determine if they believe a financial product is under or overpriced. From the well-known Black-Scholes model, which takes as input the volatility, strike and maturity of an option contract to compute the option value, it is possible to calculate the implied volatility by inversion if we know the option value, but not the volatility. This way, it is possible to define a volatility surface, thus creating a 3-dimensional plot. However, the question then arises as to how such a volatility surface moves over time and when a recalibration of the chosen model would be required.

In recent years, a wide range of techniques has been proposed to calibrate implied volatility surfaces. Nowadays, traditional methods such as Surface Stochastic Volatility Inspired (SSVI) and simple spline-based algorithms are the ones that are widely used in practice. However, the demand for more accurate and robust techniques is increasing as implied volatility shape tend to differ much over time and among different companies. As a result, scholars tend to investigate the applicability of different types of neural networks.

The aim of this research is to compare different learning techniques with respect to their ability to fit the market while adhering to financial conditions such as no-arbitrage conditions. One can also think of limit conditions that such a surface must meet. At the same time, the models need to be be linked to criteria that are important for a trader to use (or not use) a particular model, like the accuracy of the model at certain areas of a volatility curve. It is important to bring in financial conditions that have to be met in practice into a neural network to make sure that they are taken into account, while, at the same time, we require a flexible technique to represent many different implied volatility surface shapes.

The focus of this project will be how to implement these neural networks, understand them, adjust them according to investor's desires and find good metrics to compare the output of these models.

In practice, we can distinguish two types of models for an implied volatility surface: the indirect models and the direct models. The indirect models are those that depend on other dynamic models, including stochastic volatility models and L ́evy models. This type of model is not always usable. The methods we will be concerned with are the direct models, where implied volatility will be explicitly defined. Here again we can distinguish between static direct methods and dynamic direct methods. Dynamic direct methods make certain assumptions about the course of the surface over time, while static direct methods use a parametric approach.

A model that is used extensively in the industry today is one that takes the form of the so-called Stochastic Volatility Inspired (SVI) model. For a predetermined expiration, a slice is constructed using a parametric formula, the underlying parameters of which must in turn be optimized using an optimization method. A simplification of this method has been sought regarding the incorporation of no-arbitrage conditions, which led to the revelation of the Surface Stochastic Volatility Inspired, also abbreviated as SSVI, which is popular among practitioners. Non-parametric methods that use polynomials of sigmoid functions have also been used, however, these models are typically not sufficently flexible to represent the implied volatility shapes that we currently see in the market. Recently, several Machine Learning models have also been applied in the field of volatility surfaces.

To develop a robust and flexible neural network model, option data from the SP 500 index of October 2nd 2020 will be used to investigate the ability of the different models to construct volatility smiles and volatility surfaces. The SP 500 is a stock market index that tracks 500 large-cap firms in the United States. It reflects the stock market's success by monitoring the biggest corporations' risks and returns. Initially, the intention is to apply neural networks for one specific trading day in order to understand the models before expanding to a data set that covers multiple days. The original data set contains the following features: the bid price, the ask price, the strike price, the expiration and whether an option is a call or put option. A very important aspect is that we do not have a large number of strike price and maturity dates avaialbew, and, moreover, they are not ev enly distributed among the strike and time axis. We have more quotes for strike prices at the money (ATM), for example, and for short time to maturity options. The data is thus unevenly distributed, which may hamper the successful use of of-the-shelve neural networks.

Before using the data, some operations may be applied on it. First of all, the option contracts whose bid price is less than a specific value may be removed. The reason for this is that such a price comes close to the minimum price change between the bid and ask price of an option. Moreover, some inaccurate quotes may be filtered out, as they may relate to noisy data. Subsequently, a number of features, such as the log-moneyness and the mid-price, the price that lies in the middle of the bid and the ask price, are calculated and added to the data set.

We lack two main input values needed to use the Black-Scholes formula to obtain implied volatilities. These are the interest rate and the index's price. There is, however, the possibility of providing an approximation for these values by using the put-call parity. This relationship can be used to calibrate the forward price of the index, by considering call and put options with the same expiration and strike price. Since for every combination of a put option and a call option with the same expiration and strike, we not only possess the mid-price for both the put and call, but also the bid and ask prices, we could obtain different put-call differences from our data set by combining bid prices and ask prices. We will then use these put-call differences to obtain a forward price and an interest rate for each of the expirations separately. A linear regression model may be utilized to achieve this.

Certain neural network configurations which work well for one data set belonging to a trading day may not necessarily guarantee the performance of a data set belonging to another trading day. This is a robustness issue that forms an important criterion for the success of this project.

As mentioned earlier, traders may have a specific opinion of what a fit should look like to be used in practice. The part for which the log-moneyness values are around 0, should be close to perfect.

Lemma: Let

$$d_{\pm}(m,T) = -\frac{m}{\sqrt{T}\sigma_{BS}} \pm \frac{1}{2}\sqrt{T}\sigma_{BS}(m,T),$$

and let $n(\cdot)$ be the standard normal density, $N(\cdot)$ the corresponding distribution. Note that T represents the time to expiration, m stands for the log-moneyness and σ_{BS} is the implied volatility. The following conditions for the implied volatility surface should hold:

1. Positivity: For $(m, T) \in \mathbb{R} \times \mathbb{R}^+,\ \sigma_{BS}(m, T) > 0$;

2. Twice differentiability: For $T > 0$, the function $m \to \sigma_{BS}(m, T)$ is twice differentiable on $\mathbb{R}$.

3. Monotonicity: For $m \in \mathbb{R},\ T \to \sqrt{T}\sigma_{BS}(m, T)$ is increasing on $\mathbb{R}^+$, then

$$\sigma_{BS}(m, T) + 2T\frac{\partial \sigma_{BS}(m, T)}{\partial T} \geq 0.$$

4. Butterfly arbitrage-free: For $(m, T) \in \mathbb{R} \times \mathbb{R}^+$,

$$\left[1 - \frac{m\dfrac{\partial \sigma_{BS}(m,T)}{\partial m}}{\sigma_{BS}(m,T)}\right]^2 - \frac{1}{4}\left[\sigma_{BS}(m,T)T\frac{\partial \sigma_{BS}(m,T)}{\partial m}\right]^2 + T\sigma_{BS}(k,T)\frac{\partial^2 \sigma_{BS}(m,T)}{\partial m^2} \geq 0$$

5. Limit condition: If $T > 0$, then

$$\lim_{m \to \infty} d_+(m, T) = -\infty$$

6. Right boundary: If $m \geq 0$, then

$$N(d_-(m,T)) - \sqrt{T}\frac{\partial \sigma_{BS}(m,T)}{\partial m}n(d_-(m,T)) \geq 0$$

7. Left boundary: If $m < 0$, then

$$N(-d_-(m,T)) + \sqrt{T}\frac{\partial \sigma_{BS}(m,T)}{\partial m}n(d_-(m,T)) \geq 0$$

8. Asymptotic slope: For $T > 0$, $2|m| - \sigma_{BS}^2(m, T) \cdot T > 0$

The main question in this problem is can we define a neural network that can guarantee that these conditions will be satisfied, plus the fact that the ATM implied volatility is close to perfect?

Reference:

Yu Zheng, Yongxin Yang and Bowei Chen, Incorporating Prior Financial Domain Knowledge into Neural Networks for Implied Volatility Surface Prediction. see https://arxiv.org/pdf/1904.12834.pdf