Deep Finance - Models, Calibration, Hedging

Jonas - Kristian

OpenCampus

SoSe 2022 - 20. Juli 2022

Outline

From Options and Calibration To Hedging

Abbildung: Stockprice and the PayOff-Profile for a Call-Option

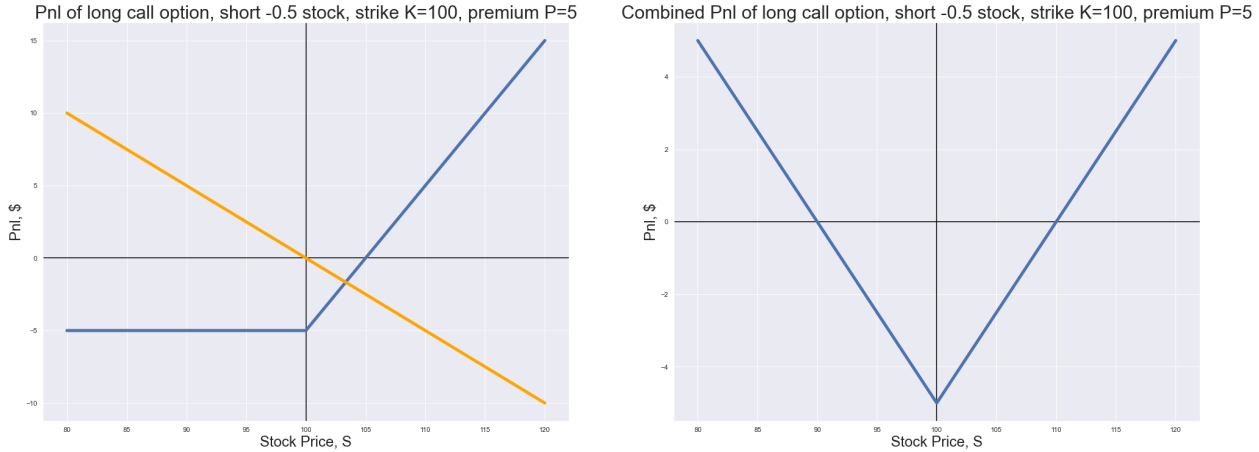


Abbildung: Profit-Loss of Combined Position

Final Goal: Improving the Hedging-Performance

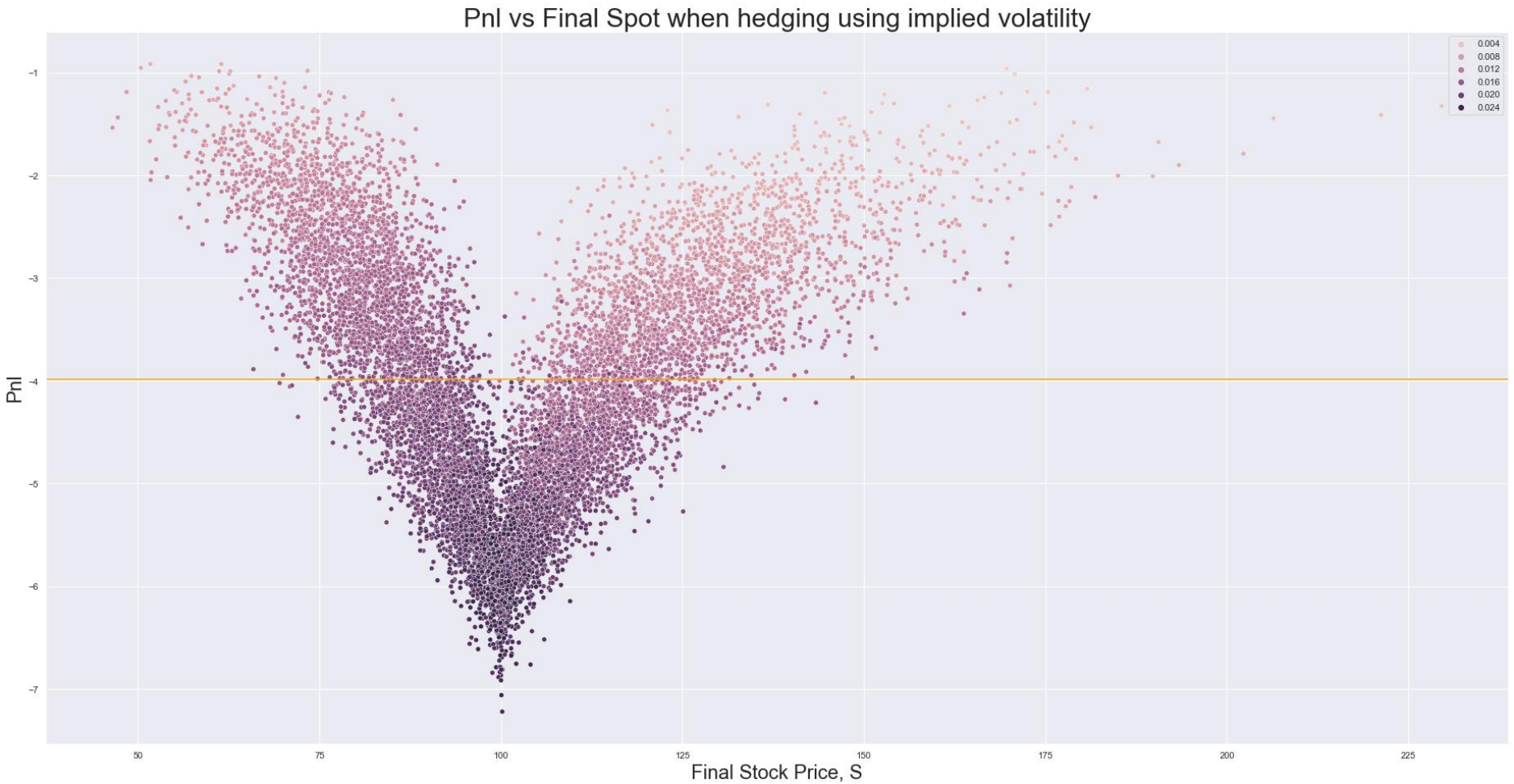


Abbildung: Profit-Loss of Hedging Position

Black-Scholes-Dynamics - Stockmarkets

$$dS_t = \mu S dt + \sigma S dW \tag{1}$$

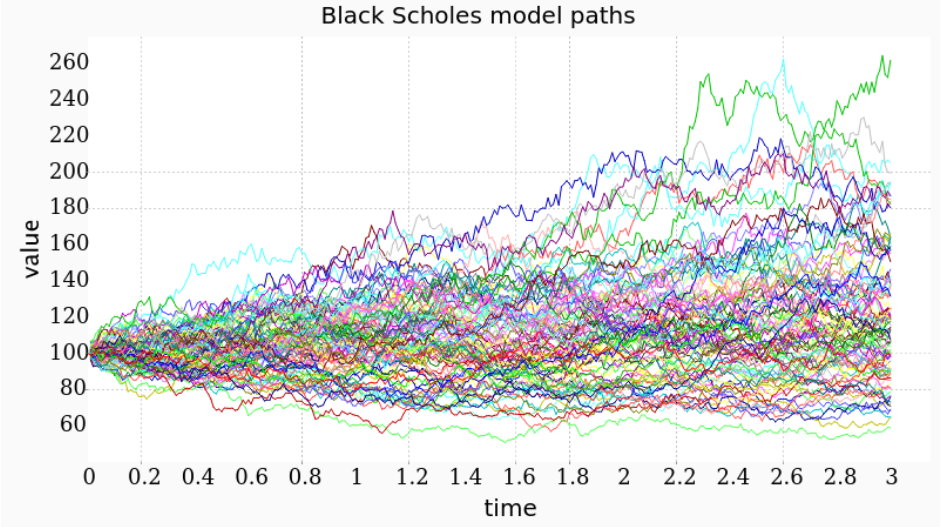


Abbildung: Black-Scholes sample paths

Vasicek-Dynamics - Interest Rate Products

$$dr_t = \alpha(\beta - r_t)dt + \sigma dW \tag{2}$$

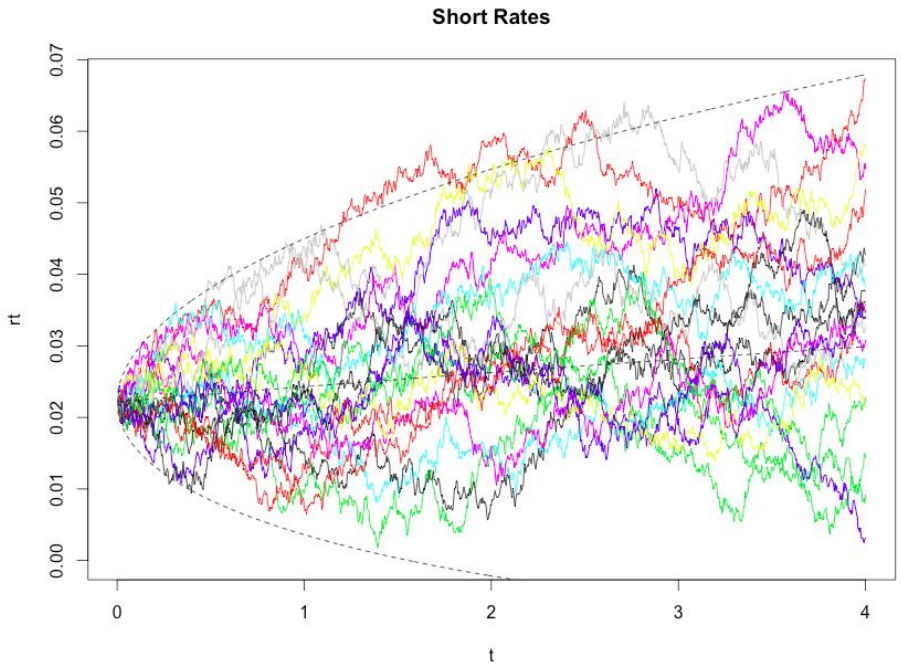


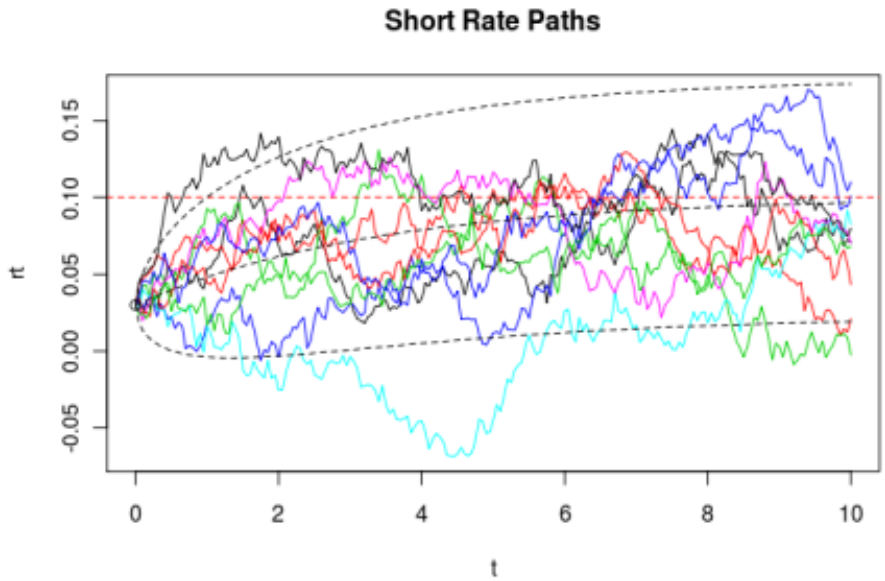
Abbildung: Black-Scholes sample paths

Heston Dynamics - Advanced Model

$$dS_t = \mu S_t dt + \sqrt{\nu_t} S_t dW_t^S \tag{3}$$

$$d\nu_t = \kappa(\theta - \nu_t)dt + \xi\sqrt{\nu_t} dW_t^\nu \tag{4}$$

$$dW_t^S W_t^\nu = \rho dt \tag{5}$$



Outline

Black-Scholes Formula - Call Price

$$Call = C(S,t) = S\Phi(d_1) - Ke^{-r(T-t)}\Phi(d_2) \tag{6}$$

$$d_1 = \frac{ln(S/K) + (r + \sigma^2/2)(T-t)}{\sigma\sqrt{T-t}} \tag{7}$$

$$d_2 = \frac{ln(S/K) + (r - \sigma^2/2)(T-t)}{\sigma\sqrt{T-t}} \tag{8}$$

$$\Phi(x) = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}} exp(\frac{-z^2}{2})dz \tag{9}$$

$$Delta = \Delta = \frac{dC}{dS} = \Phi(d_1); Vega = \frac{dC}{d\sigma} \tag{10}$$

Real Market Data - SP500

Marktbericht > S&P 500

3.825,33

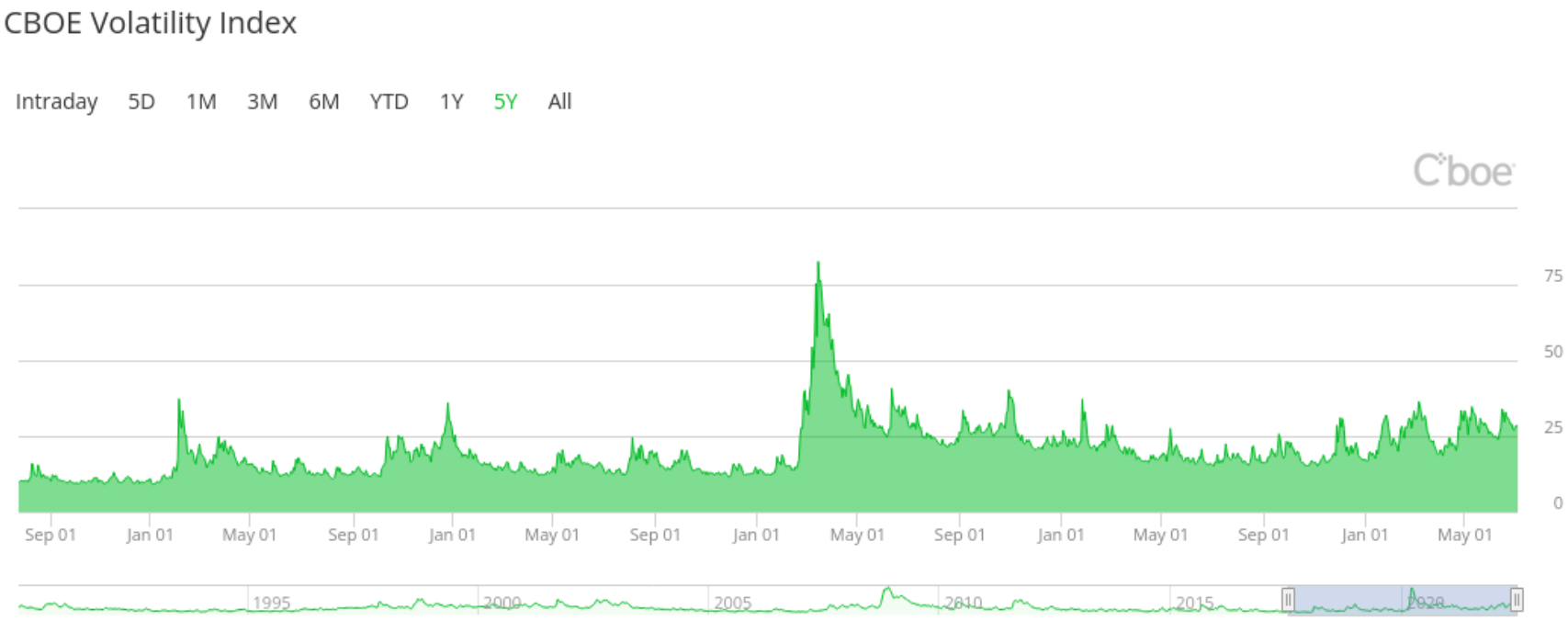
+1.400,15 (57,73 %) ↑ in den letzten 5 Jahren

1. Juli, 17:45 GMT-4 • Haftungsausschluss

1 T. | 5 T. | 1 M. | 6 M. | YTD | 1 J. | 5 J. | Max.

Eröffnung	3.781,00	Tief	3.752,10	52-Wo-Hoch	4.818,62
Hoch	3.829,82	Vort.|Schl.	3.785,38	52-Wo-Tief	3.636,87

Real Market Data - VIX



Real Market Data - Reality

Volatility Surface Calibration

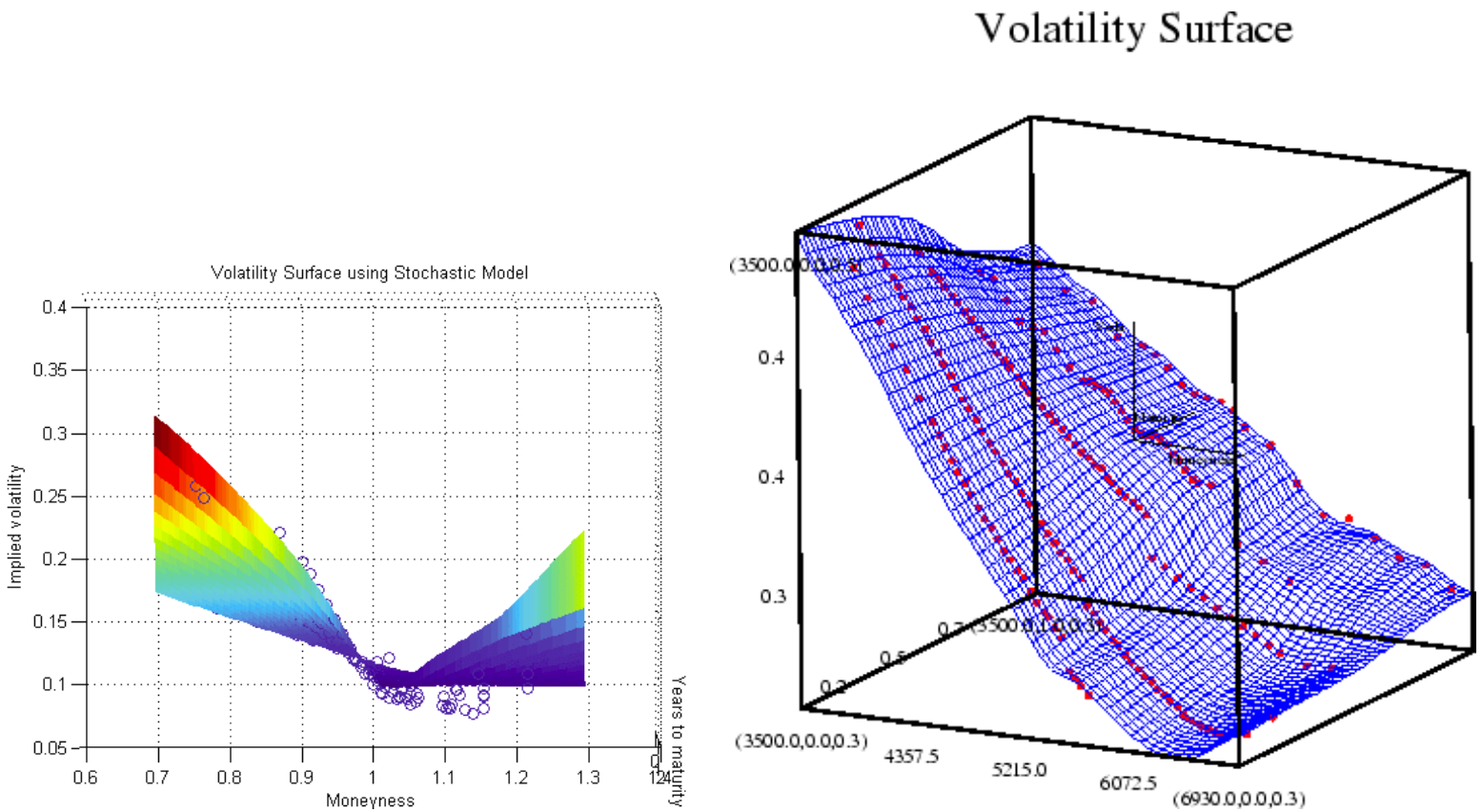


Abbildung: From data-points to the volatility-surface

Volatility Surface and Smile

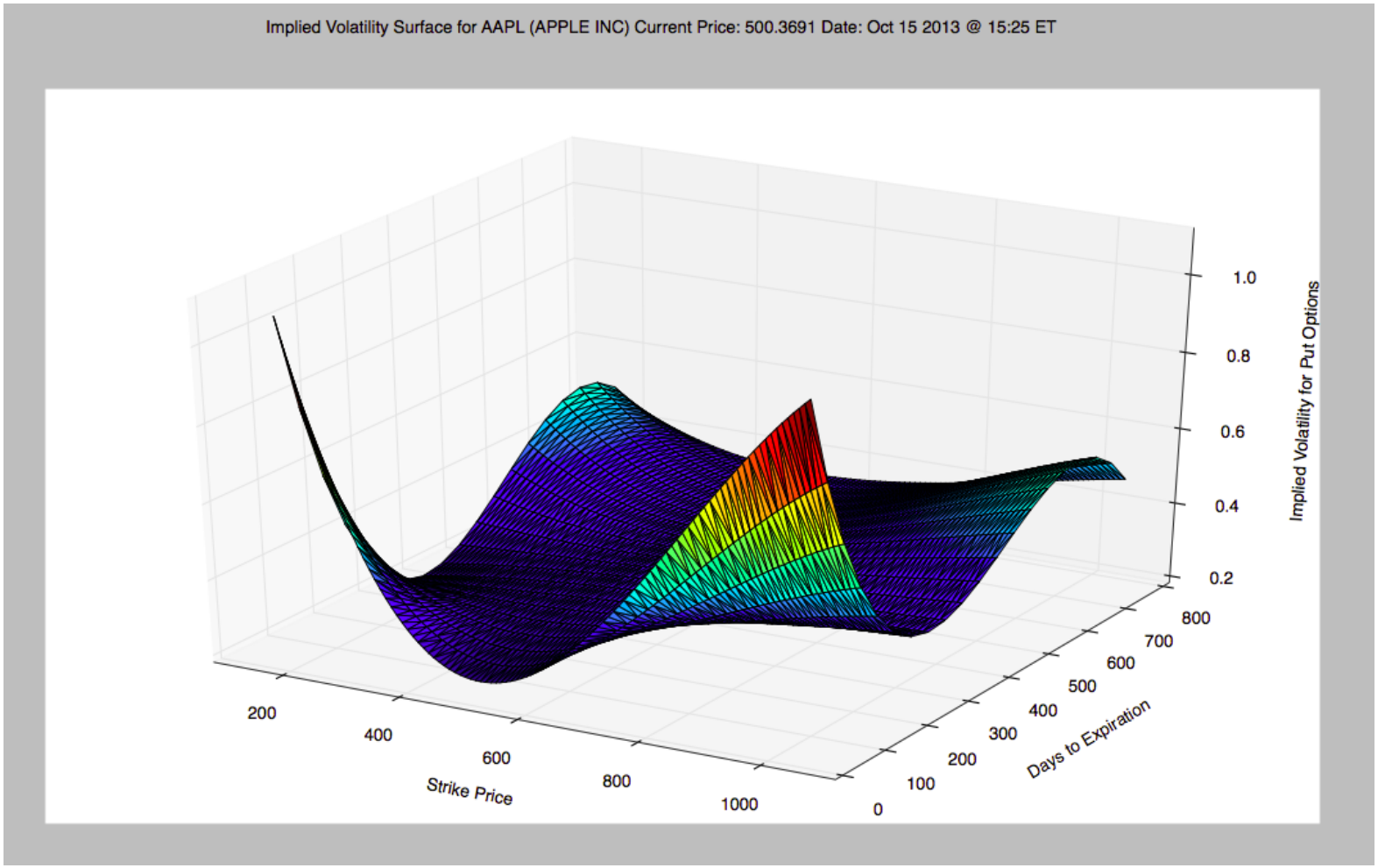


Abbildung: Implied Volatility Surface for Apple on Oct. 15 2013

Volatility Surface Calibration Procedure

The Implied Volatility is that value σ that must be inserted into the Black-Scholes (BS) formula in order to retrieve the option price quoted in the market:

$$BS\left(S, K, T, r, \sigma\right) = P \tag{11}$$

where S is the underlying spot price, K is the strike, T time to maturity, r risk free interest rate and the option price quoted in the market. All these quantities are observable. The only non observable quantity is σ.
How to find the value such that for fixed S, K, T, r

$$BS\left(S, K, T, r, \sigma\right) - P = 0 \tag{12}$$

We have to use numerical methods or NEURAL NETWORKS !!!

Outline

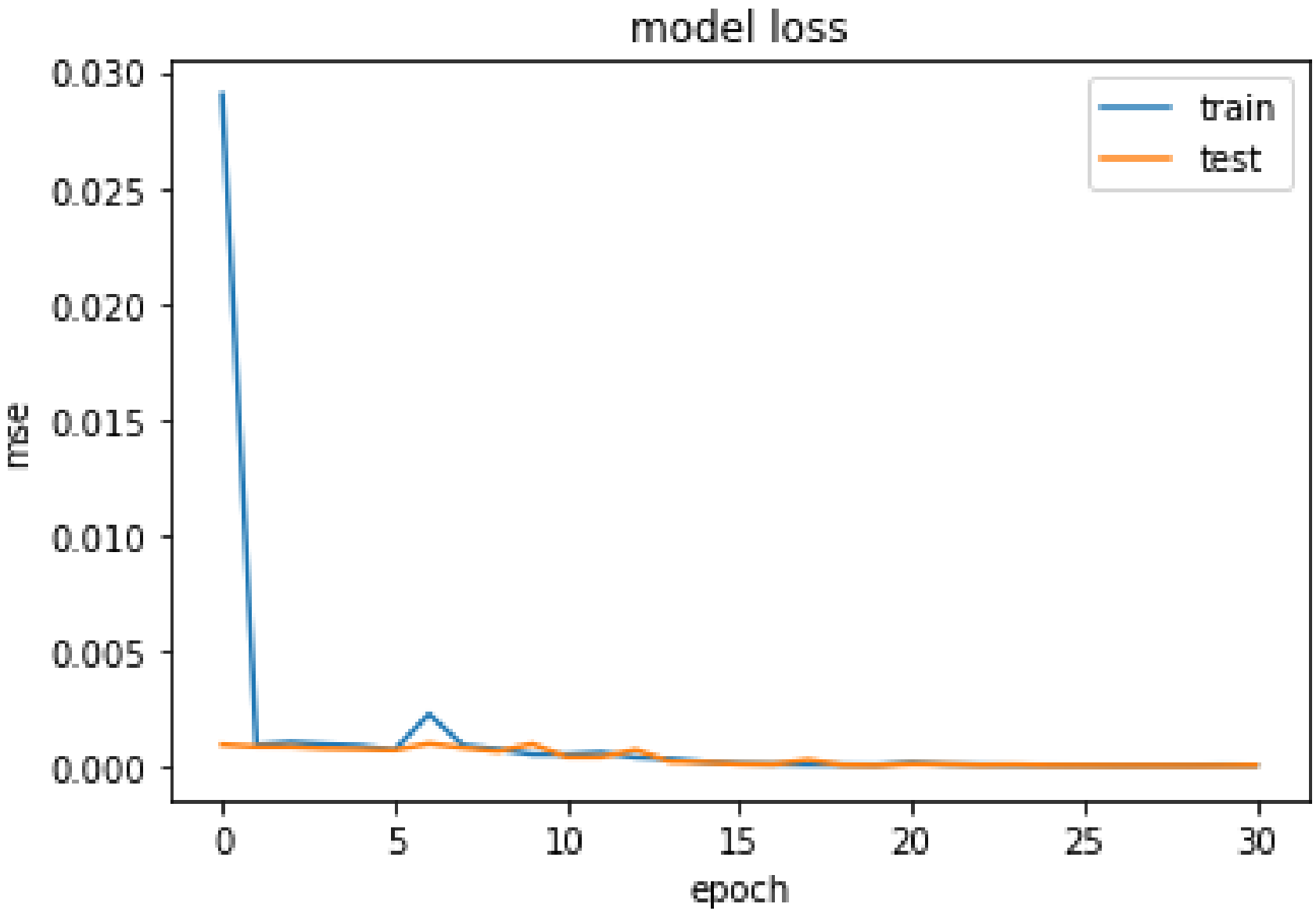
Network Layout

```
# Netz bauen, noch kein finetuning
from keras.layers import Activation
model = tf.keras.models.Sequential([
    #tf.keras.layers.InputLayer(input_shape = (5,)), # Anzahl Features
    tf.keras.layers.Dense(units = 200, activation = 'relu', input_shape = (5,)),
    #tf.keras.layers.Dropout(rate = 0.25),
    tf.keras.layers.Dense(units = 200, activation = 'relu'),
    #tf.keras.layers.Dropout(rate = 0.25),
    tf.keras.layers.Dense(units = 200, activation = 'relu'),
    #tf.keras.layers.Dropout(rate = 0.25),
    tf.keras.layers.Dense(units = 200, activation = 'relu'),
    #tf.keras.layers.Dropout(rate = 0.25),
    tf.keras.layers.Dense(units = 1, activation = 'linear')
])
model.compile(loss='mse',optimizer='adam')
# Model configuration: Liu et al. (2019)
```

Callback

```python
from tensorflow.keras.callbacks import EarlyStopping
callback = EarlyStopping(monitor='loss', patience=4)

#tf.keras.callbacks.EarlyStopping(monitor='val_loss',
#                                 min_delta=0,
#                                 patience=0,
#                                 verbose=0,
#                                 mode='auto',
#                                 baseline=None,
#                                 restore_best_weights=False)

# https://blog.paperspace.com/tensorflow-callbacks/
```

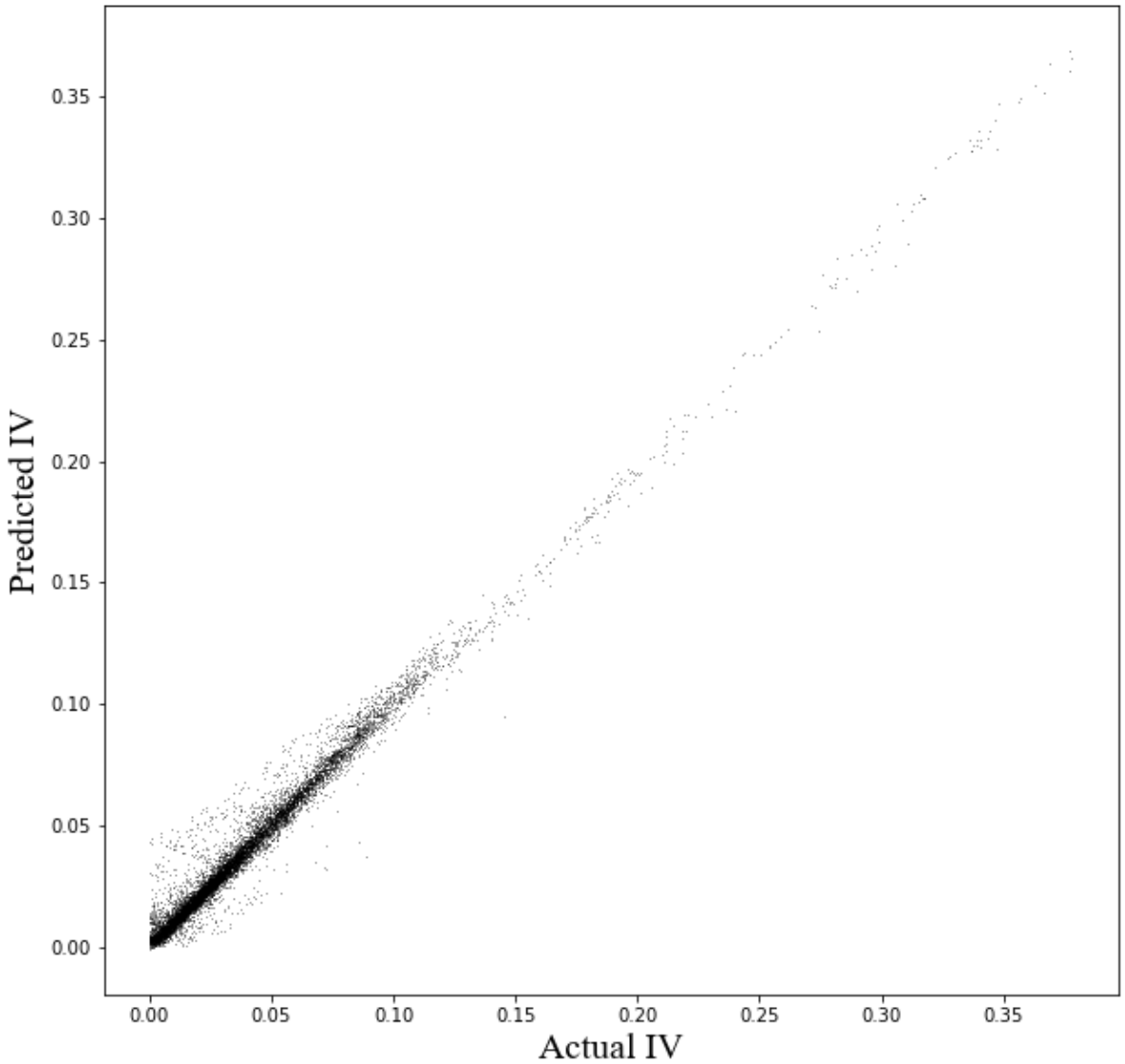
Training Process



Performance on the test data

	stock_price	strike_price	maturity	moneyness	pred_iv	imp_vol	mae
954765	56	36	1.333333	0.642857	0.015065	0.015307	0.000242
711377	52	26	1.750000	0.500000	0.024962	0.027945	0.002982
1086810	58	53	1.666667	0.913793	0.023278	0.025432	0.002154
859684	54	63	1.000000	1.166667	0.028662	0.025687	0.002975
1040050	57	68	0.500000	1.192982	0.072568	0.069612	0.002955
...	...	...	...	...	...	...	...
530703	49	21	1.583333	0.428571	0.006716	0.008757	0.002041
608920	50	44	1.833333	0.880000	0.016793	0.014334	0.002459
591292	50	23	1.833333	0.460000	0.015252	0.011389	0.003862
462411	47	80	1.083333	1.702128	0.058949	0.057476	0.001473
1120191	59	23	1.166667	0.389831	0.003196	0.002344	0.000852

Performance on the test data



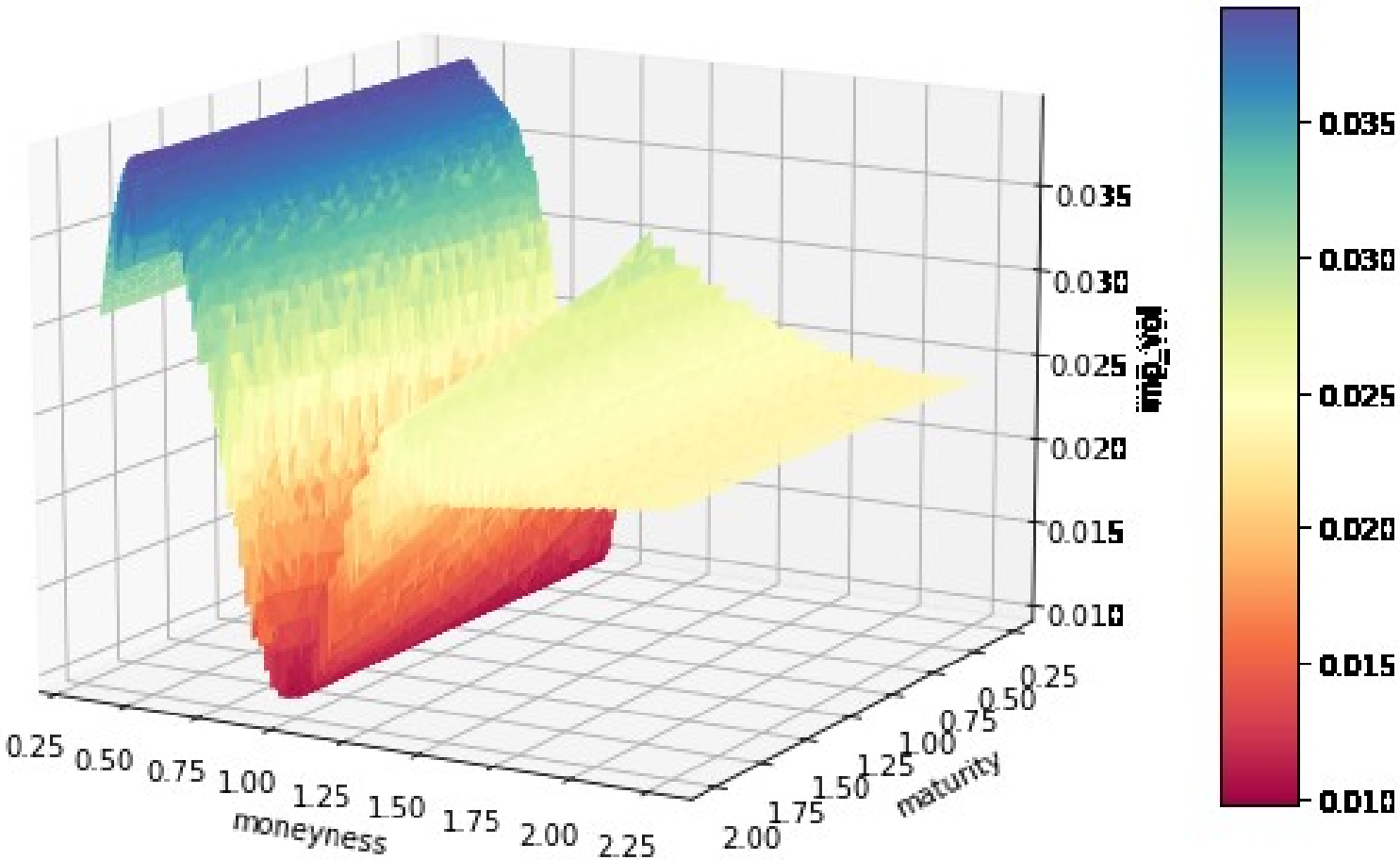
Plotting the implied volatility surface

	stock_price	strike_price	maturity	risk_free_rate	sigma	black_scholes	option_price	imp_vol
0	40.0	20.0	0.2500	0.01	0.1	NaN	NaN	NaN
1	40.0	20.0	0.3375	0.01	0.1	NaN	NaN	NaN
2	40.0	20.0	0.4250	0.01	0.1	NaN	NaN	NaN
3	40.0	20.0	0.5125	0.01	0.1	NaN	NaN	NaN
4	40.0	20.0	0.6000	0.01	0.1	NaN	NaN	NaN
...	...	...	...	...	...	...	...	...
9256	60.0	90.0	1.6500	0.01	0.1	NaN	NaN	NaN
9257	60.0	90.0	1.7375	0.01	0.1	NaN	NaN	NaN
9258	60.0	90.0	1.8250	0.01	0.1	NaN	NaN	NaN
9259	60.0	90.0	1.9125	0.01	0.1	NaN	NaN	NaN
9260	60.0	90.0	2.0000	0.01	0.1	NaN	NaN	NaN

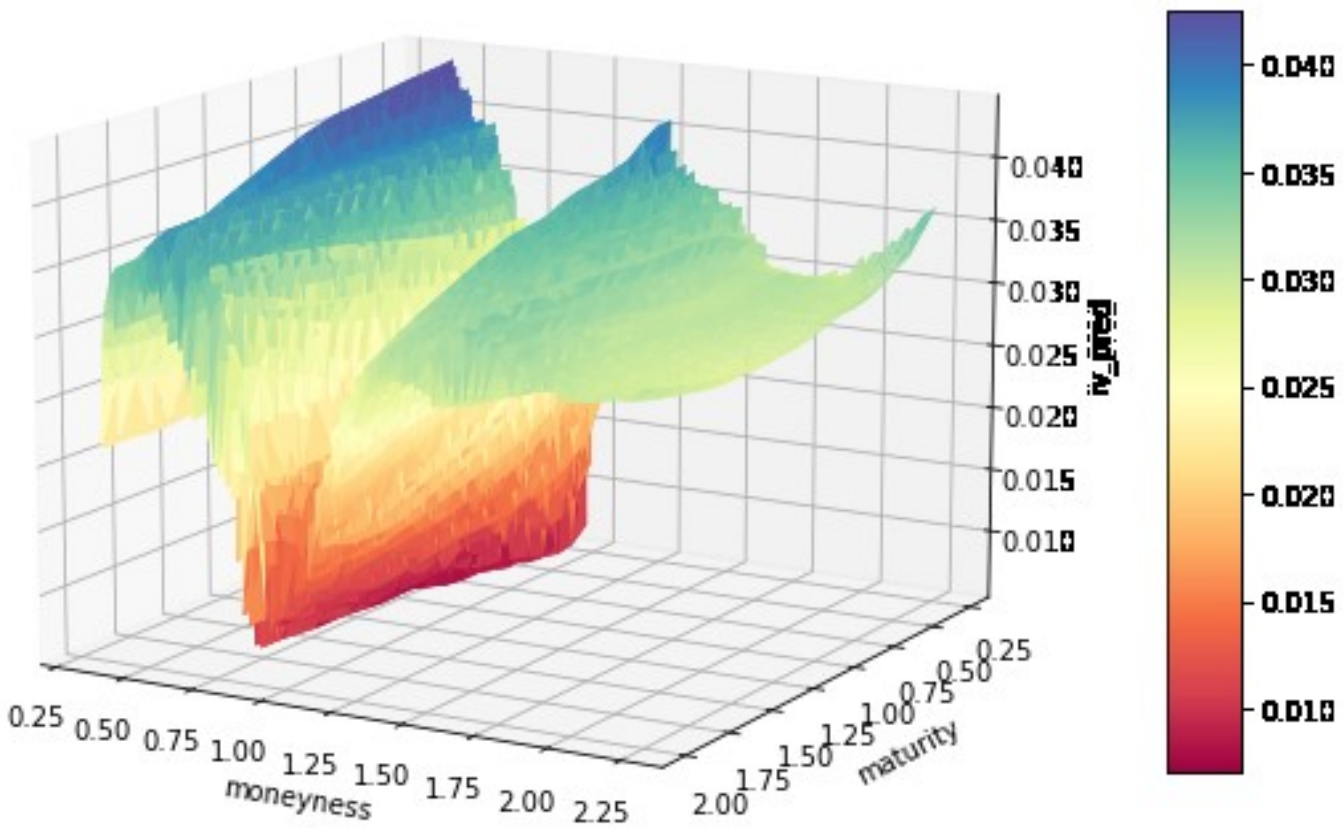
Plotting the implied volatility surface

	stock_price	strike_price	maturity	risk_free_rate	sigma	black_scholes	option_price	imp_vol	iv_pred	moneyness	mae
0	40.0	20.0	0.2500	0.01	0.1	20.049938	21.069326	0.038066	0.038913	0.5	0.000847
1	40.0	20.0	0.3375	0.01	0.1	20.067386	21.086775	0.038095	0.038788	0.5	0.000693
2	40.0	20.0	0.4250	0.01	0.1	20.084820	21.104208	0.038123	0.038662	0.5	0.000539
3	40.0	20.0	0.5125	0.01	0.1	20.102238	21.121626	0.038150	0.038536	0.5	0.000386
4	40.0	20.0	0.6000	0.01	0.1	20.119641	21.139029	0.038178	0.038520	0.5	0.000343
...	...	...	...	...	...	...	...	...	...	...	...
9256	60.0	90.0	1.6500	0.01	0.1	0.003233	1.022621	0.026244	0.020849	1.5	0.005395
9257	60.0	90.0	1.7375	0.01	0.1	0.004454	1.023843	0.026215	0.021253	1.5	0.004962
9258	60.0	90.0	1.8250	0.01	0.1	0.005970	1.025358	0.026179	0.021550	1.5	0.004629
9259	60.0	90.0	1.9125	0.01	0.1	0.007810	1.027199	0.026135	0.021839	1.5	0.004296
9260	60.0	90.0	2.0000	0.01	0.1	0.010006	1.029394	0.026083	0.021973	1.5	0.004109

Volatility Surface for synthetic data



Learned Volatility Surface for synthetic data



Outline

Trading Strategy

The table below gives the history of the share price of Range Ltd. over the past 12 months, together with the value and delta of a call maturing at the end of the 12-month period. At the beginning of the period you purchase 10,000 calls which you delta-hedge immediately and then on a monthly basis. Compute your monthly and cumulative P&L, including your purchase cost and option payoff. Assume zero interest rates.

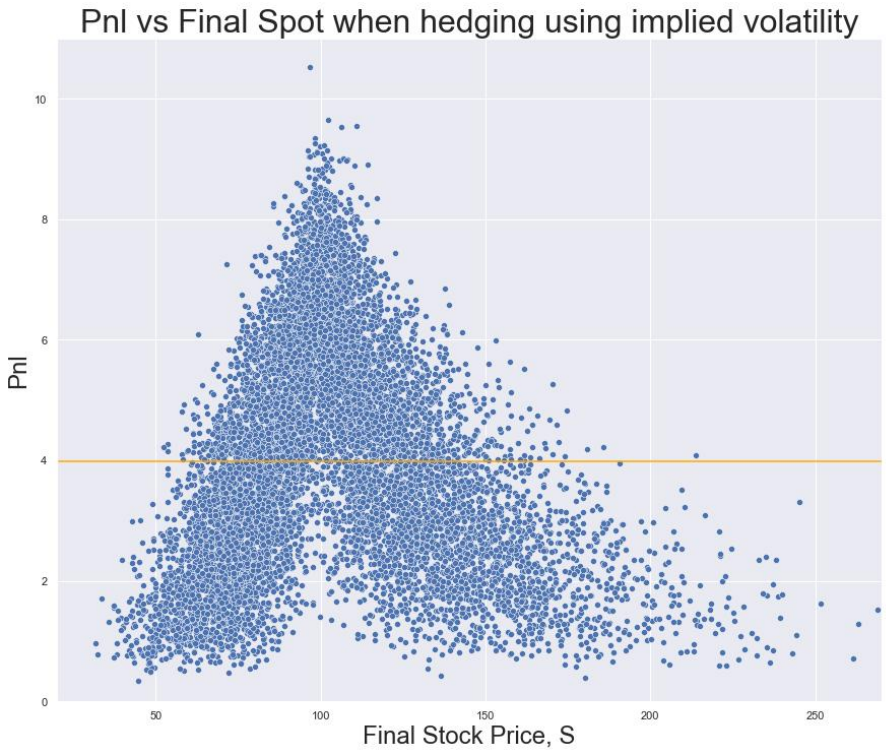
Month	Range Ltd Price (£)	Call Value (£)	Delta (per £)
0	100	11.84	0.61
1	90	6.04	0.43
2	105	13.89	0.68
3	90	5.02	0.41
4	85	2.8	0.29
5	95	6.06	0.48
6	100	8.01	0.58
7	110	14.07	0.78
8	115	17.46	0.87
9	125	26.16	0.97
10	120	20.8	0.97
11	115	15.4	0.98
12	110	10	1

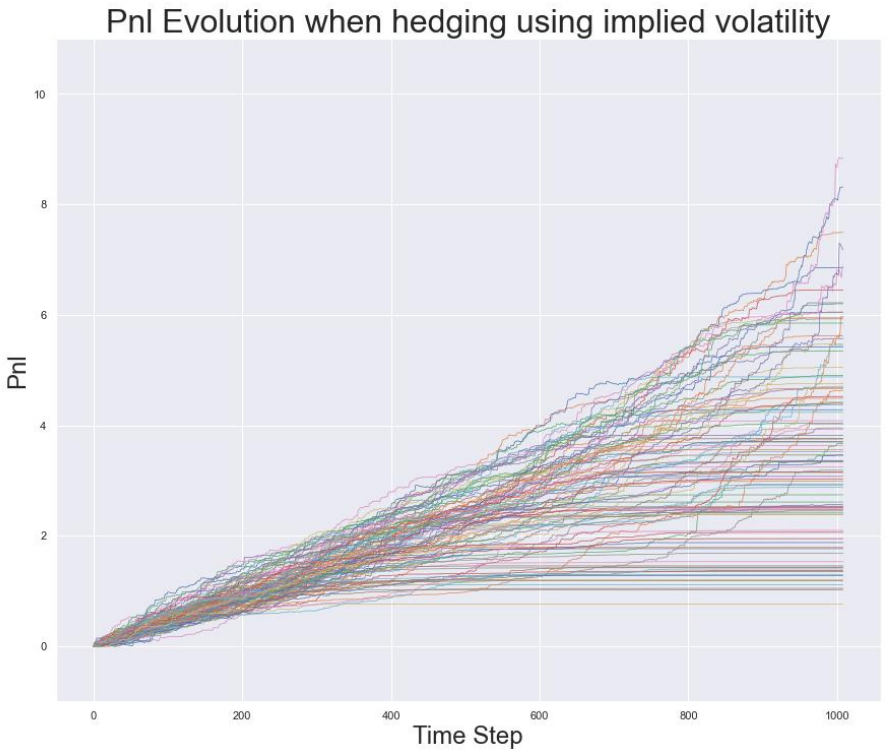
Trading Strategy

Month	Range Ltd Price (£)	Call Value (£)	Delta (per £)	"Dollar" Gamma	Stock position*	Monthly P&L** (£)	Cumulative P&L (£)
0	100	11.84	0.61	76.61	−6,100	−118,400†	−118,400
1	90	6.04	0.43	73.96	−4,300	3,000	−115,400
2	105	13.89	0.68	82.02	−6,800	14,000	−101,400
3	90	5.02	0.41	80.57	−4,100	13,300	−88,100
4	85	2.8	0.29	70.87	−2,900	−1,700	−89,800
5	95	6.06	0.48	99.12	−4,800	3,600	−86,200
6	100	8.01	0.58	110.57	−5,800	−4,500	−90,700
7	110	14.07	0.78	100.73	−7,800	2,600	−88,100
8	115	17.46	0.87	83.66	−8,700	−5,100	−93,200
9	125	26.16	0.97	31.12	−9,700	0	−93,200
10	120	20.8	0.97	38.37	−9,700	−5,100	−98,300
11	115	15.4	0.98	41.42	−9,800	−5,500	−103,800
12	110	10	1	0		95,000‡	−8,800

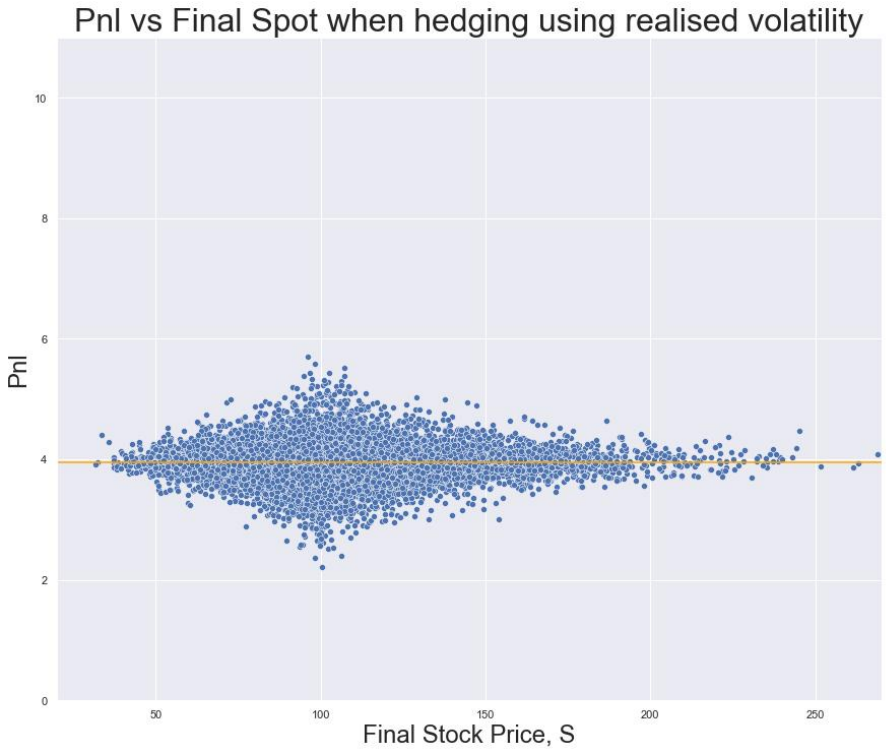
*Stock position: Number of shares at the end of the month.
**Monthly P&L: 10,000 times the change in call value, plus the previous number of shares times the change in stock price.
†Purchase cost.
‡Includes the call payoff of £100,000.

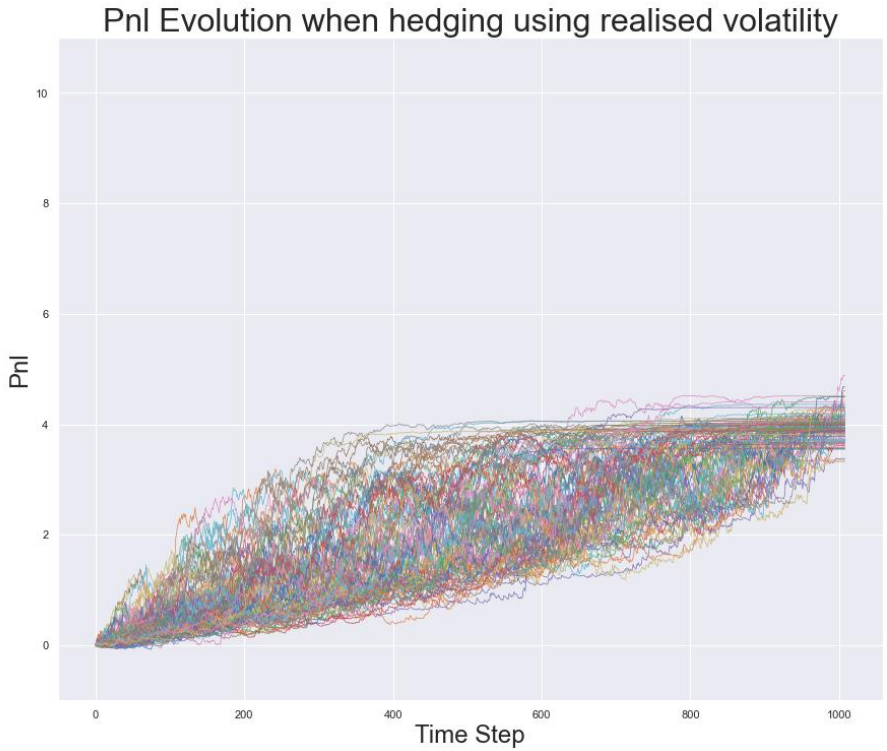
Hedging Error for implied volatility



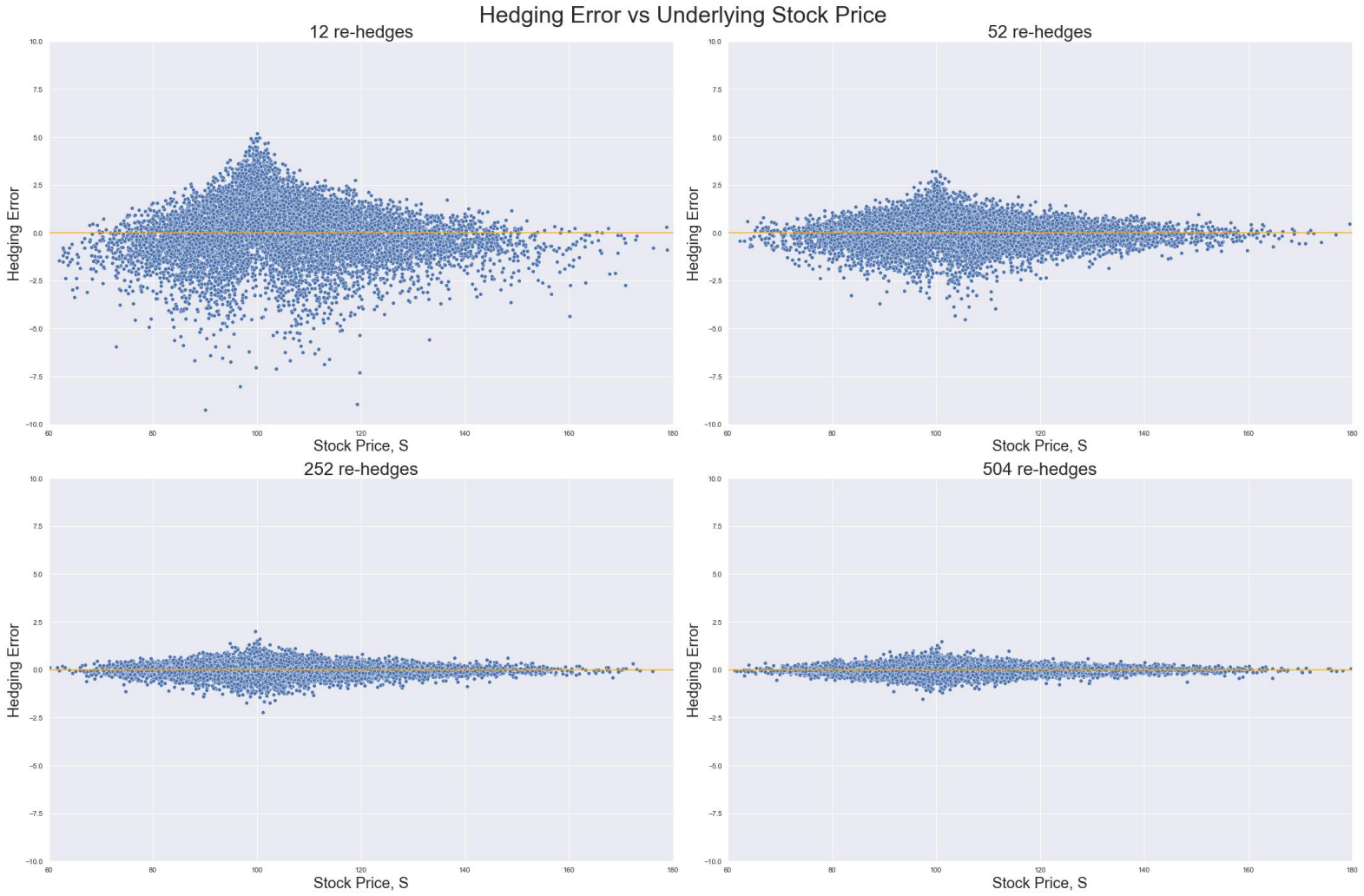


Hedging Error for realized volatility





Hedging Error vs. Re-Hedging



Deep Hedging - Still to come

Quadratic Hedging Criterion:

$$\sum_{(K,T)} E\left[\left((S_T - K)_+ - C_{\text{market}}(K,T) - (H^{(K,T)} \bullet S)_t\right)^2\right] \to \min! \quad (13)$$

In this setting we choose that the input of each neural network in the implementation depends only on the current price, i.e. $H_t = g_t(S_t)$ and g_t denotes a neural network.
We can view the above as a supervised learning problem: the input data x_i correspond to trajectories of S_t, the output y_i should be 0 and the loss function is given by

$$\mathcal{L} = \frac{1}{K} \sum_i \left(f(S_T(\omega_i)) - \pi - \sum_{i=0}^{N-1} g_i(S_i(\omega_i))(S_{i+1}(\omega_i) - S_i(\omega_i)) \right)^2 . \quad (14)$$

Extracting weights from NN

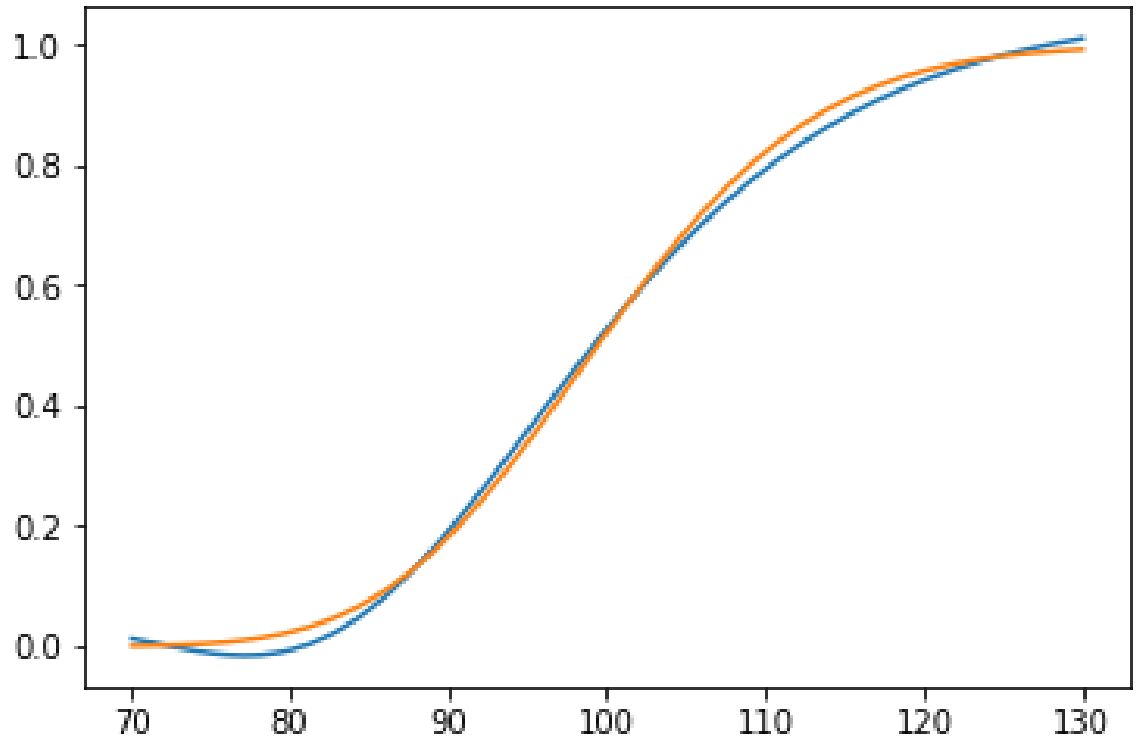
```
weights = model_hedge.get_weights()
#print(weights)
```

```
#This works when the number of layers equals d=2

def deltastrategy(s,j):
    length=s.shape[0]
    g=np.zeros(length)
    for p in range(length):
        ghelper=np.tanh(s[p]*(weights[j*2*d])+weights[j*2*d+1])
        #for k in range(1,d-1): #this line has to be checked in the case for d >2
        #    ghelper=np.tanh(np.matmul(weights[2*k+j*2*d], ghelper[0])+weights[2*k+j*2*d+1])
        g[p]=np.sum(np.squeeze(weights[2*(d-1)+j*2*d])*np.squeeze(ghelper))
        g[p]=g[p]+weights[2*d-1+j*2*d]
    return g
```

```
s=np.linspace(70,130,60) # range for the asset price to compute the hedging strategy
k=21 # choose k between 1 and N-1

learneddelta=deltastrategy(s,k)
truedelta=scipy.norm.cdf((np.log(s/strike)+0.5*(T-k*T/N)*sigma**2)/(np.sqrt(T-k*T/N)*sigma))
```

Learned Delta vs. True Delta

```
# This plots the true versus the learned Hedging strategy
plt.plot(s,learneddelta,s,truedelta)
plt.show()
```



Literaturverzeichnis I

Liu et al.
A neural network-based framework for financial model calibration.
4 2019.
https://arxiv.org/abs/1904.10523.

Teichmann et al.
Machine learning in finance - spring 2022.
ETH Zürich.

Hafner Franke, Härdle.
Statistics of Financial Markets - An Introduction, volume 5.
Springer, 2019.

Andres Hernandez.
Model calibration with neural networks.
https://papers.ssrn.com/sol3/papers.cfm?abstract;$d = 2812140$.

Literaturverzeichnis II

Mark Jamison.
How to delta hedge an option: Part-I to part-5.
https://medium.datadriveninvestor.com/how-to-delta-hedge-an-option-part-i-2efc91b24400.

Aaron Schlegel.
Black scholes formula and python implementation.
https://aaronschlegel.me/black-scholes-formula-python.html.

Subir Varma and Sanjiv Das.
Deep learning, chapter 11 - option pricing.
https://srdas.github.io/DLBook/DeepLearningWithPython.htmloption-pricing.

Literaturverzeichnis III

Ruf Wang.
Neural networks for option pricing and hedging: A literature review.
https://papers.ssrn.com/sol3/papers.cfm?abstract;d = 3322085.

Shiwen XIA.
Calibrating rough volatility models with deep learning.
https://github.com/svenhsia/Calibrating-Rough-Volatility-Models-with-Deep-Learning.