Schätzung von impliziten Volatilitäten mit neuronalen Netzen

Thomas Weber

SciComp Inc.
Weber & Partner GbR

pailab event, 5 März 2020

SciComp

About SciComp Inc.

SciComp entwickelt Softwarelösungen sowie Produkte für die Bewertung von Finanzderivaten und für das Risikomanagement

```
der[V,t] + 1/2 sigma^2 S^2 der[V,{S,2}]
         + (r-q) S der[V,S] - r V = 0;
Payoff[Max[K-S,0]];
Output[V, S=Spot];
```

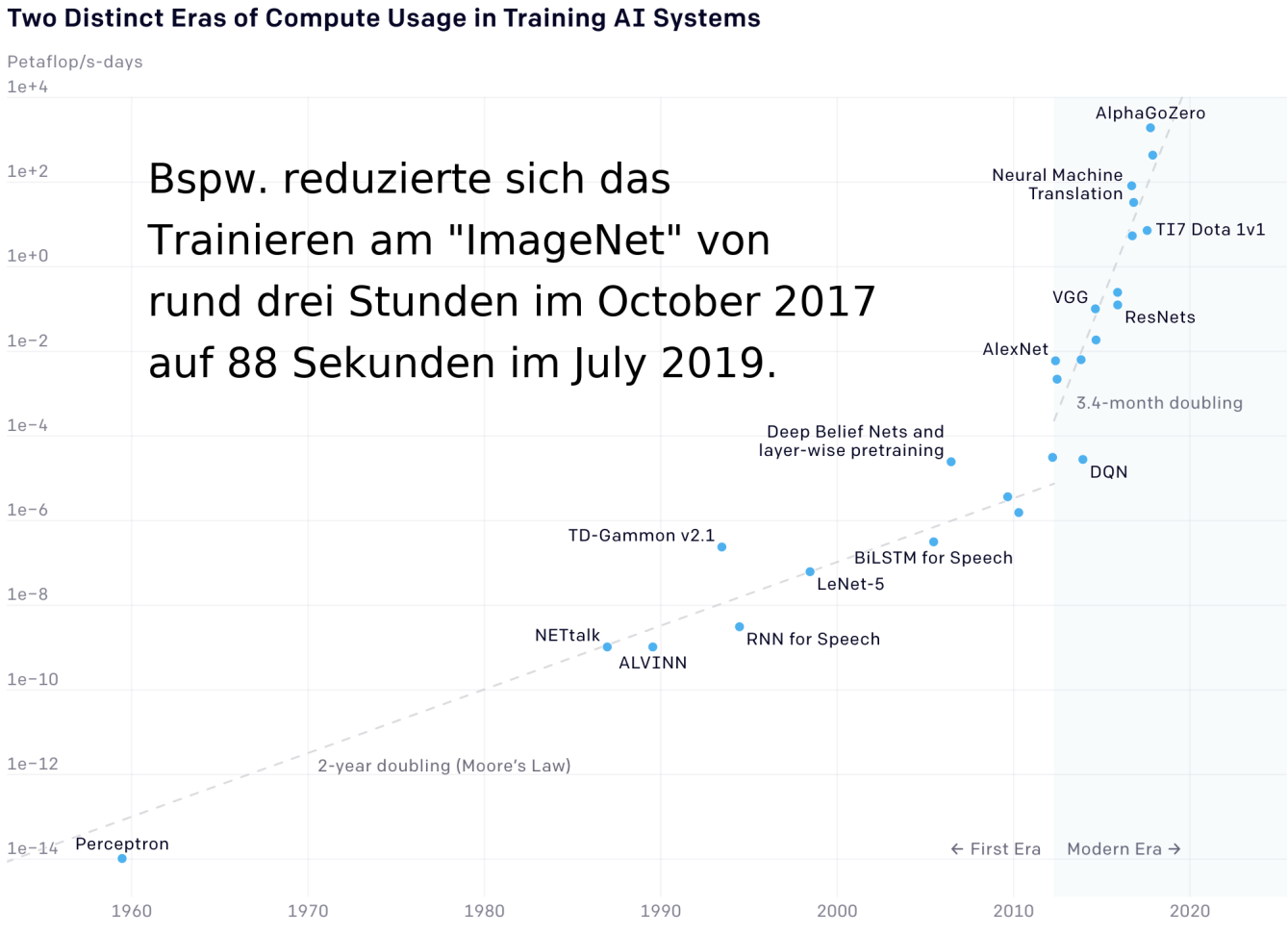
Price model for an european call option

SciComp ist führend in der Entwicklung und Weiterentwicklung von *Software synthesis engines*

SciComp

Das Training von AI Algorithmen wird schneller und günstiger

Trends



Stanford University: Artificial Intelligence Index Report 2019

SciComp

AI Algorithmen sind breit verfügbar

Trends

SciComp

AI im Investmentbanking?

Trends

Github, Februar 2020

arXiv/SSRN, Februar 2020

- ...
- Bewertung
- Hedging
- Exposureberechnung
- Kalibrierung

SciComp

Bewertung

SciComp SABR Examples

- European futures options
  - Train to implied volatility, deflated via approximate formula
  - Single hidden layer ANN, $\mathcal{O}(10^5)$ training data
  - RMS Implied vol and PV errors of 1bp, and 0.5bp respectively
  - 140,000 evaluations/sec
- Double no-touch futures option
  - Parameter space divided into regions based on estimated PV
  - Train to PV spread over Black-Scholes
  - Combined single hidden layer ANNs
  - RMS PV error of 3bp
  - 140,000 evaluations/sec

SciComp

Kalibrierung

Kalibierung ist das Fitten von nicht beobachtbaren Parametern eines Modells zu einem Datensatz

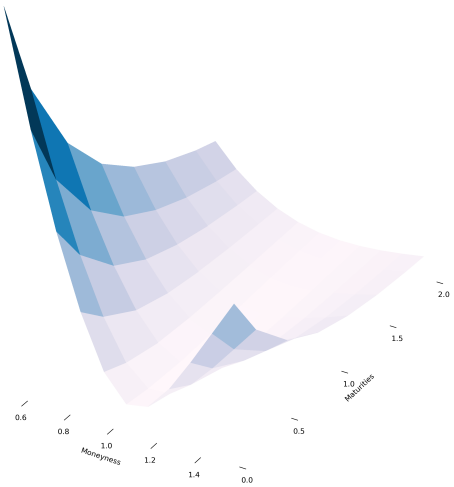
$$\hat{\theta} = \underset{\theta \in \Theta}{\operatorname{argmin}} \sum (\text{Model}(\theta) - \text{Market}(\theta))^2$$

Beispiel: Parameter für das Gausssche 2 Faktormodell für Zinsderivate

$$\theta = \{\kappa_1, \eta_1, \kappa_2, \eta_2, \rho\} = \begin{cases} dx(t) & = -\kappa_1 x_1(t) + \eta_1 dW_1 \\ & \quad -\kappa_2 x_2(t) + \eta_2 dW_2 \\ \rho & = dw_1 dW_2 \\ r(t) & = F(t) + \phi(t, \ldots) + x_1(t) + x_2(t) \end{cases}$$

SciComp

Klassische Vorgehensweise

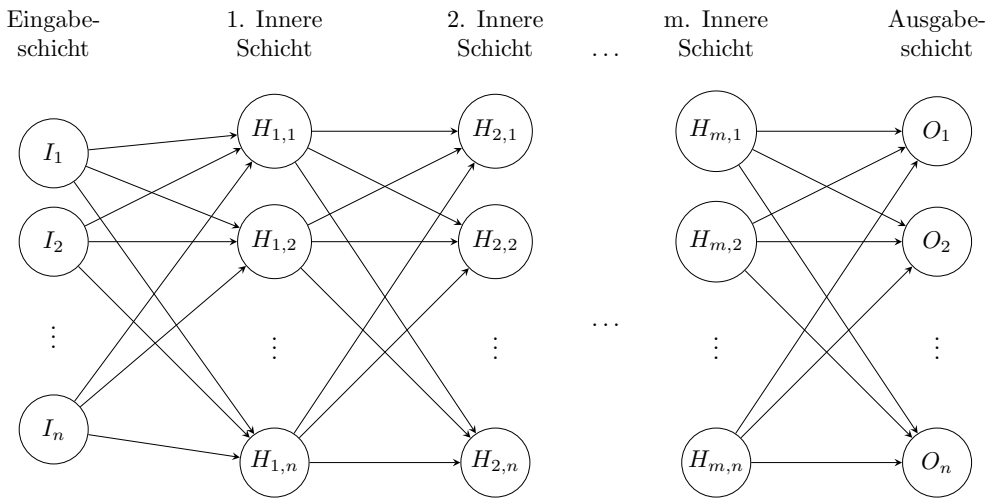


1. Marktdaten (Black-Scholes-Volatilitäten) als Eingabewerte
2. Datenbereinigung
3. Anwendung eines numerischen Suchverfahrens (z.B. Levenberg-Marquart)
4. Event. visuelle Überprüfung
5. Nutzen der gefundenen Parameter für die Bewertung von (exotischen) Derivaten

SciComp

Kalibrierung mit Neuronalen Netzen

Kalibierung ist das Fitten von nicht beobachtbaren Parametern (Knoten) eines Modells (Netzwerk) zu einem Datensatz



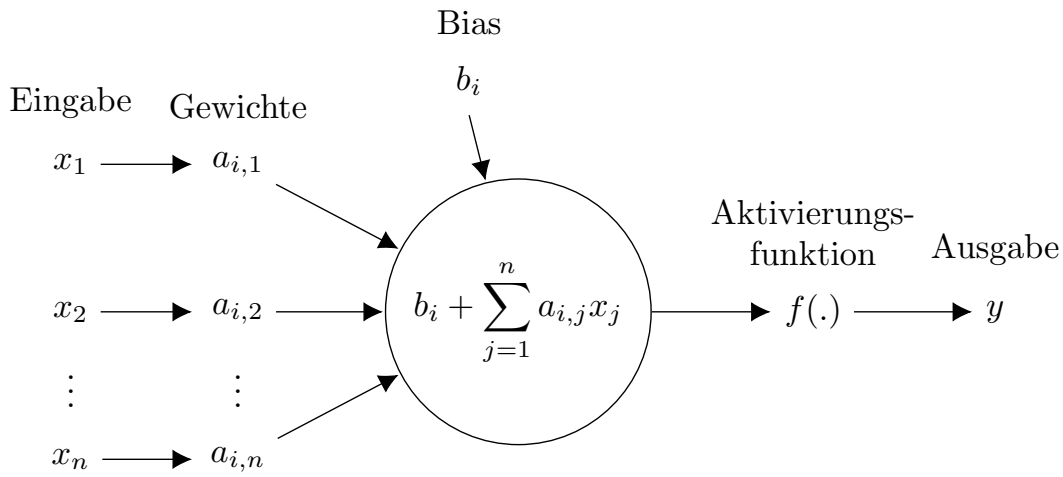
Mögliche Vorgehensweisen:

1. Direkte Kalibrierung an Marktdaten (Inverse Map)
2. In zwei Schritten
   - i Lerne ein Bewertungsmodell
   - ii Kalibriere das Bewertungsmodell anhand von Marktdaten

SciComp

Ziele des Trainierens von Neuronalen Netzen

1. Den Fehler beim Training so klein wie möglich halten (Underfitting)[1]



Einzelnes Neuron und seine Komponenten

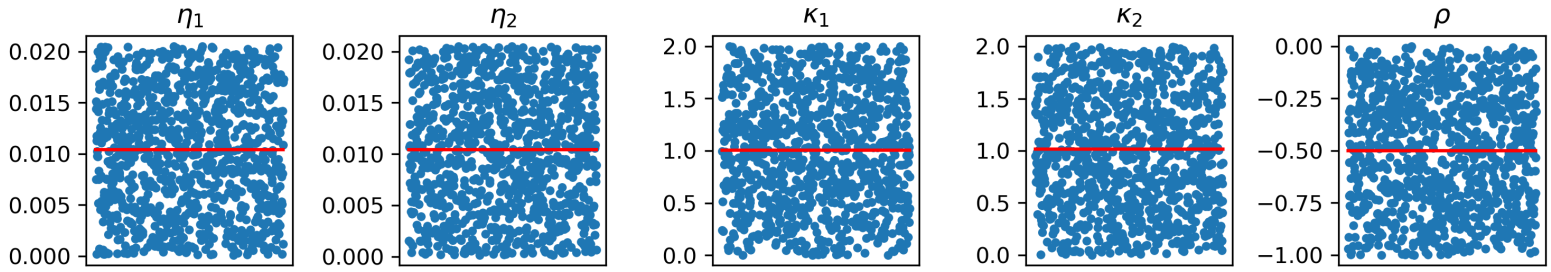
2. Den Verallgemeinerungsfehler so klein wie möglich zu halten (Overfitting)

[1] Erfolg ist garantiert! Dank des Universal Approximation Theorems, Vgl. Hull, Neural Network Approach to Understanding Implied Volatility Movement, 2019

Datenvorbereitung

Beispiel: Kalibrierung des Gaussschen 2-Faktormodells

- 50000 Datensätze mit analytischer Formel für die Bewertung von Swaptions mit unterschiedlichen Laufzeiten, Tenors und Moneyness, d.h. 1,8 Mio. Bewertungen mit gewürfelten Parametern



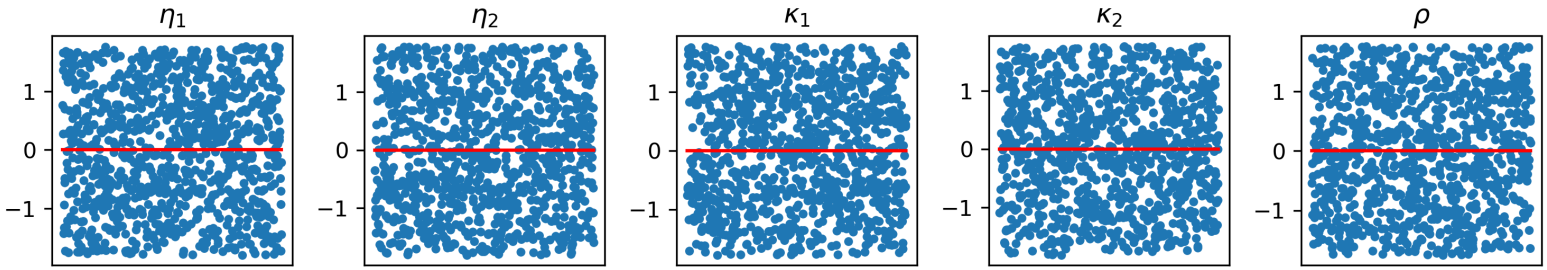
- Umrechnung in Black-Scholes Volatilitäten
- Bereinigung der Daten
  - Entfernen aller Datensätze bei der die meisten Bewertungen null sind
  - Setzen der verbleibenden 0-Bewertungen auf einen Minimalwert
- Aufteilung der Daten in Trainings- und Testdaten (15%)

SciComp

Datenskalierung und -normalisierung

Beispiel: Kalibrierung des Gaussschen 2-Faktormodells

- Verschiedene ML-Algorithmen erwarten Werte mit Durchschnitt 0 und vergleichbarer Standardabweichung
- Standard ist $z = \frac{x-\mu}{s}$, wobei μ der Durchschnitt und s die Standardabweichung aus den Trainingsdaten ist.

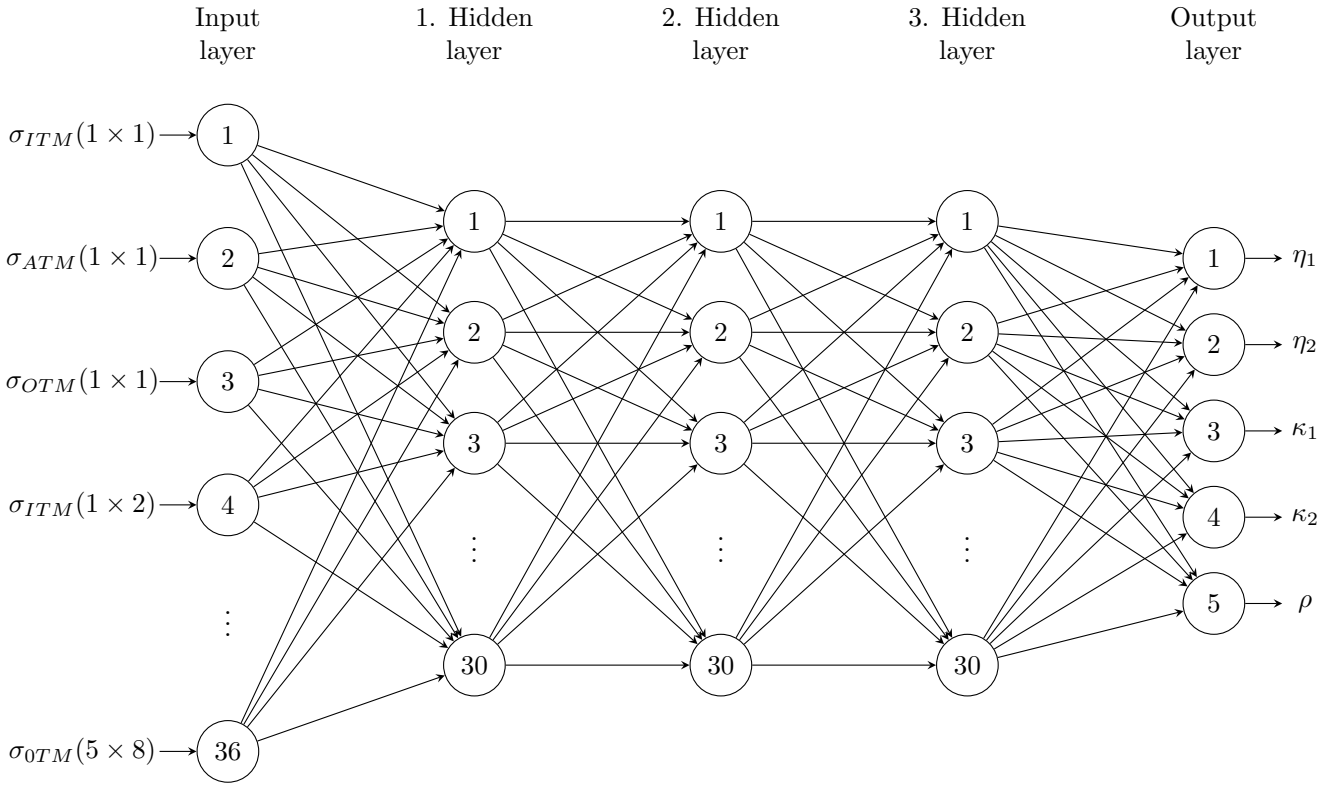


- Viele andere Möglichkeiten
- Skalierung hat Effekt auf das Ergebnis

SciComp

Kalibrierung des Gaussschen 2-Faktormodells

1. Versuch



Neural Network Forward Architektur mit 3 versteckten Schichten und 30 Neuronen in jeder versteckten Schicht, mit 36 (Markt-)Volatilitäten als Eingabe den Modelparametern (hier Gaussche 2 Faktor Model) als Ausgabe (Supervised Learning). Hier insgesamt 3125 Knoten/Parameter

SciComp

Erstellen und Trainieren des Neuronalen Netzes

1. Versuch

- Hier Verwendung von Tensorflow in Python

```
from tensorflow import keras
model = keras.models.Sequential([
        keras.Input(shape=(36)),
        keras.layers.Dense(30,activation='elu'),
        keras.layers.Dense(30,activation='elu'),
        keras.layers.Dense(30,activation='elu'),
        keras.layers.Dense(5,activation='linear')])
```

- Festlegung der Zielfunktion ...

```
model.compile(loss=root_mean_squared_error,
      optimizer='adam', metrics=['accuracy'])
```

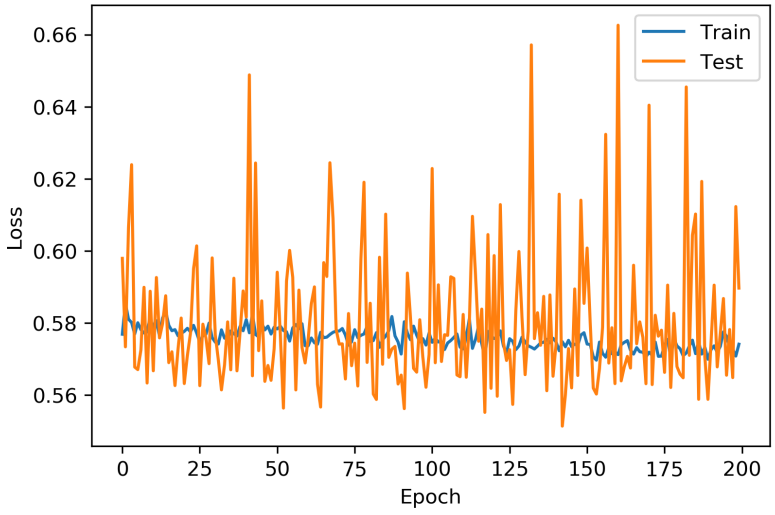
- ... Starten

```
history=model.fit(x_train_transform, y_train_transform,
    validation_data = (x_test_transform,y_test_transform)
    batch_size=32, epochs = 200, verbose = 0,shuffle=1)
```

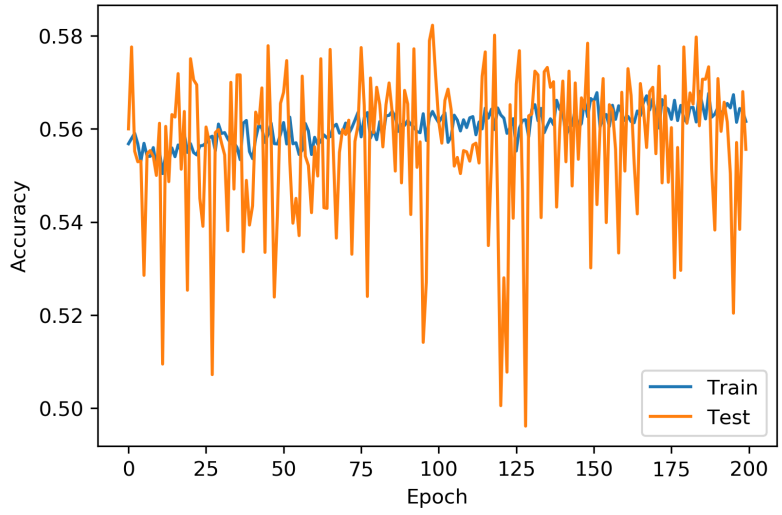
SciComp

Ergebnisse

1. Versuch



Gesamtfehler

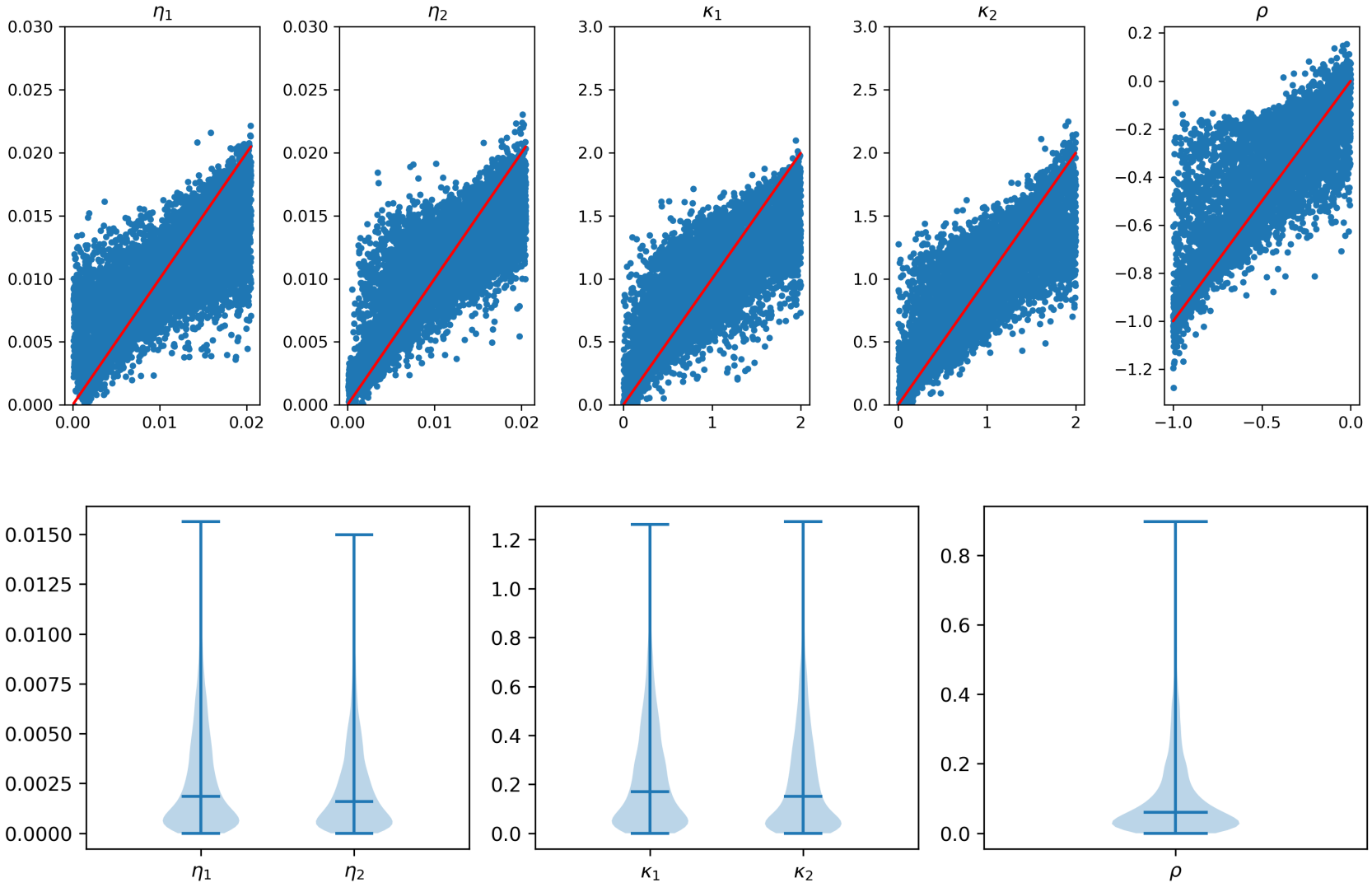


Accuracy

SciComp

Fehler bei den Parametern

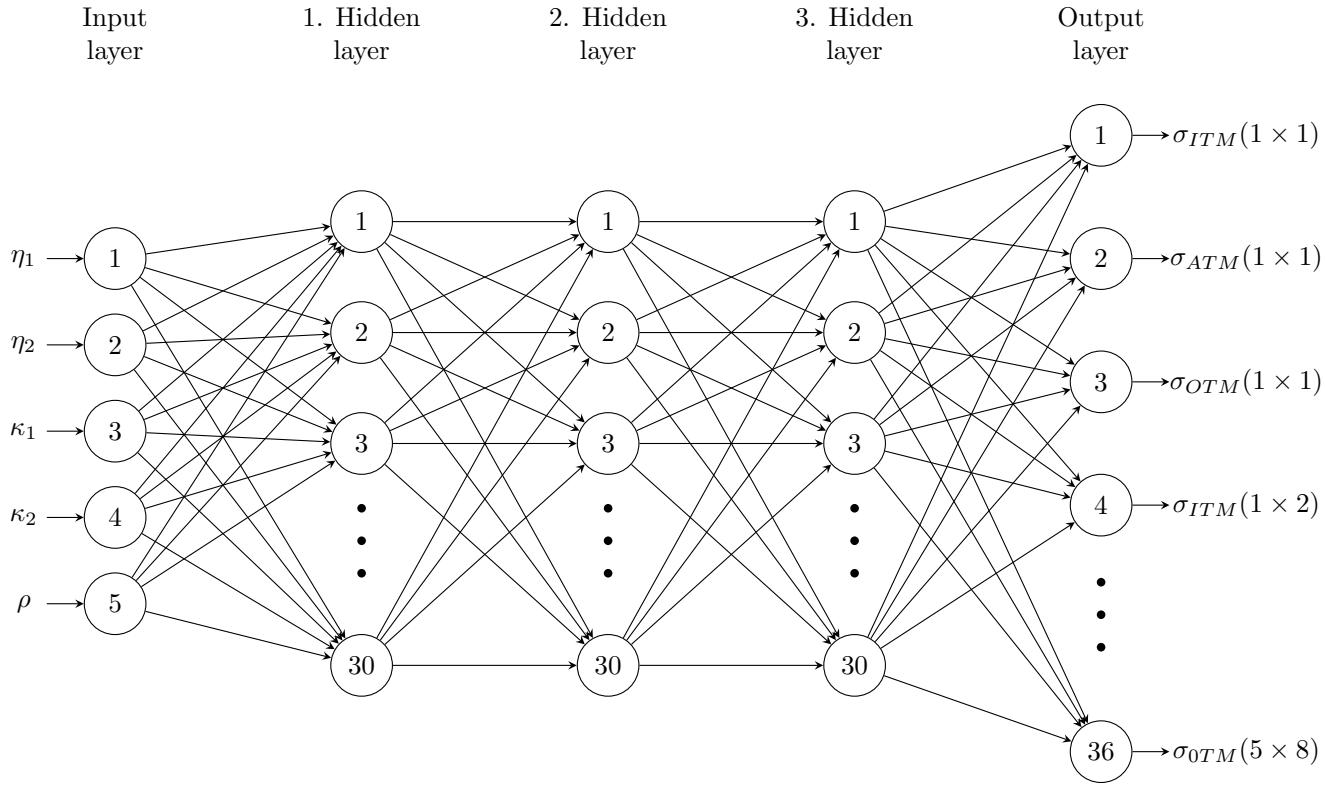
1. Versuch



Geschätzte Parameter

Kalibrierung des Gaussschen 2-Faktormodells

2. Versuch

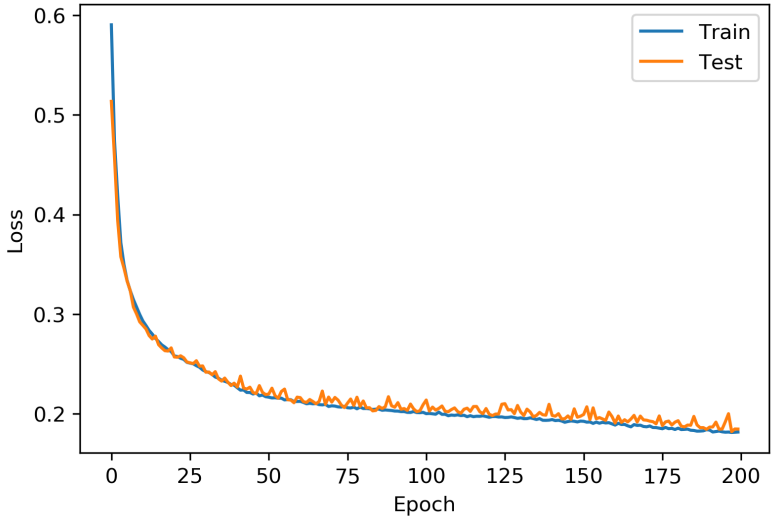


Neural Network Forward Architektur mit fünf Parameter als Eingabe und 36 MArktvolatilitäten als Ausgabe. Hier insgesamt 3156 Knoten/Parameter

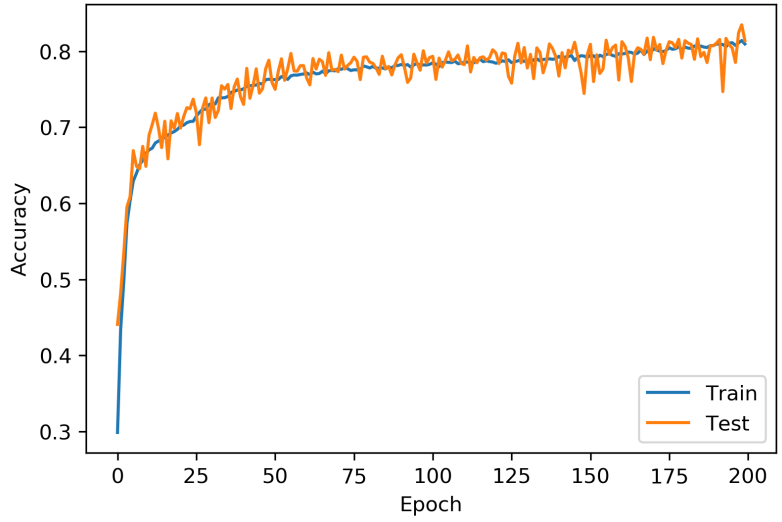
SciComp

Ergebnisse

2. Versuch



Gesamtfehler

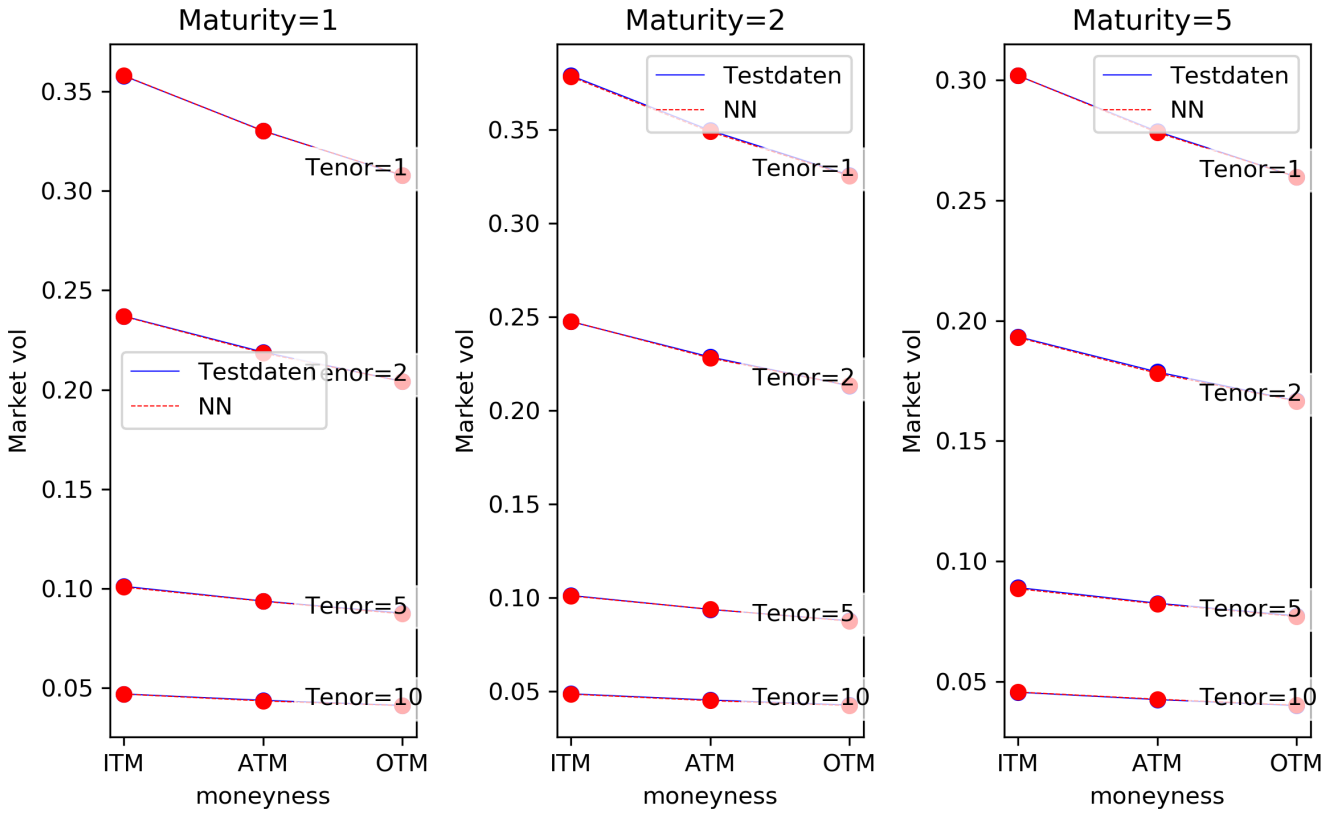


Accuracy

SciComp

Ergebnisse pro Swaption

2. Versuch

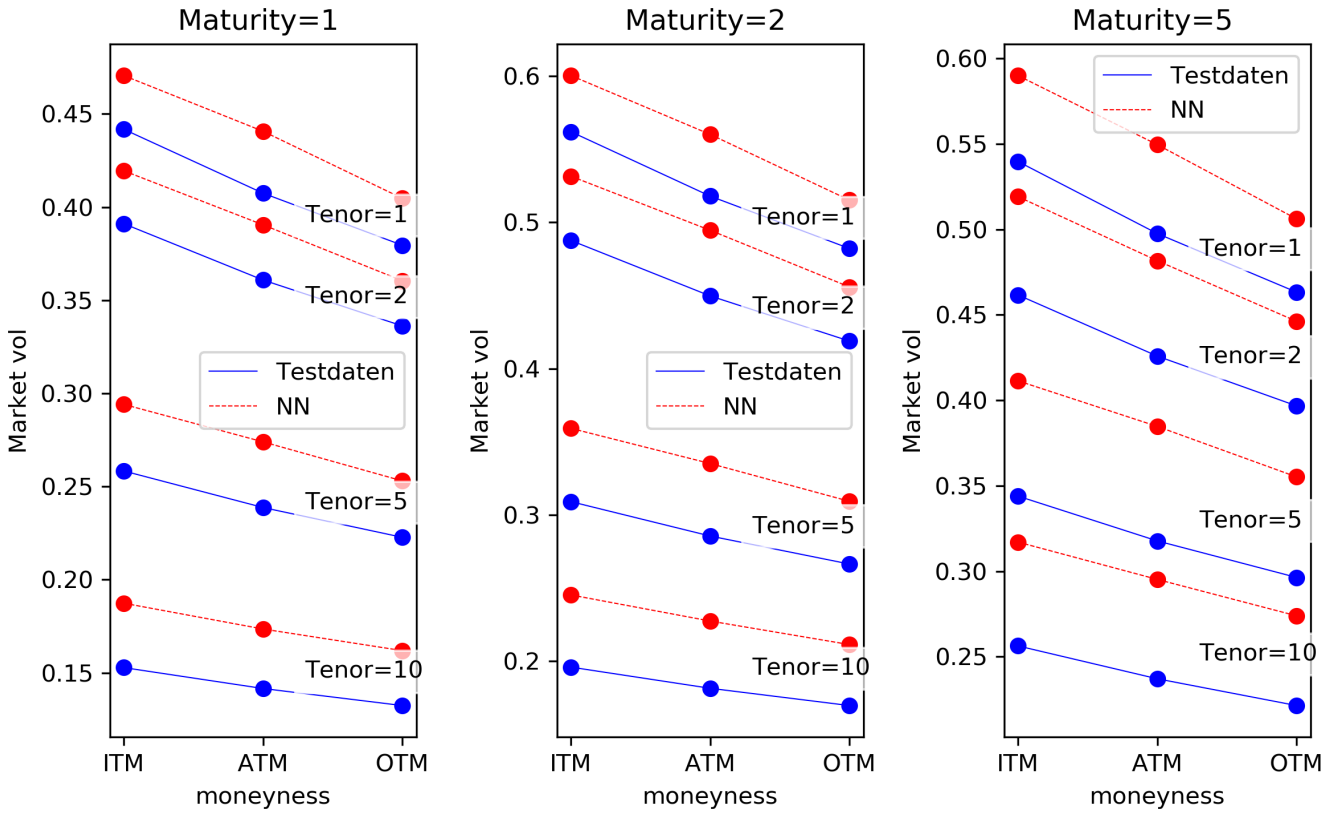


Testfall 2955

SciComp

Ergebnisse pro Swaption

2. Versuch

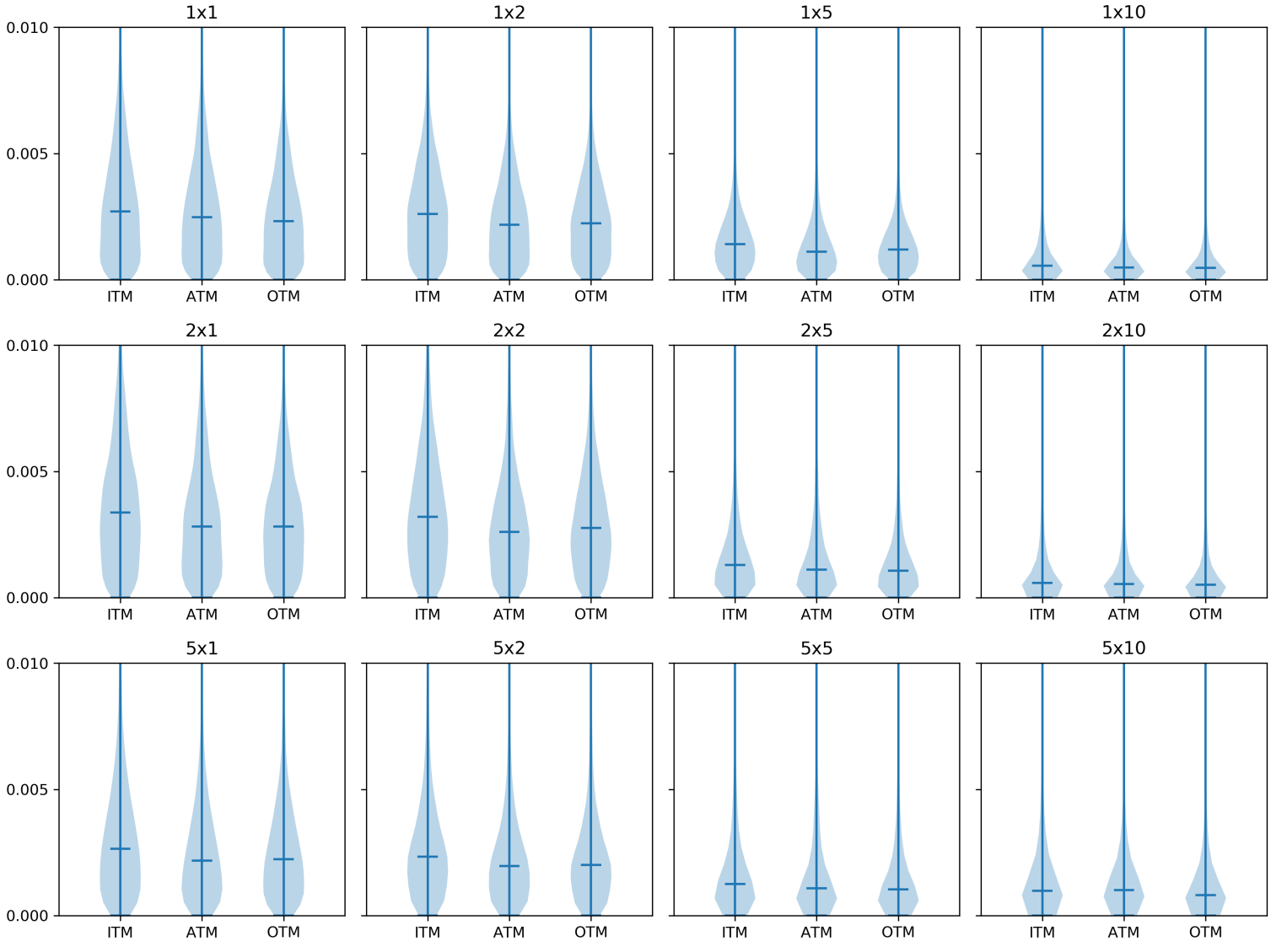


Testfall 893

SciComp

Ergebnisse pro Swaption

2. Versuch

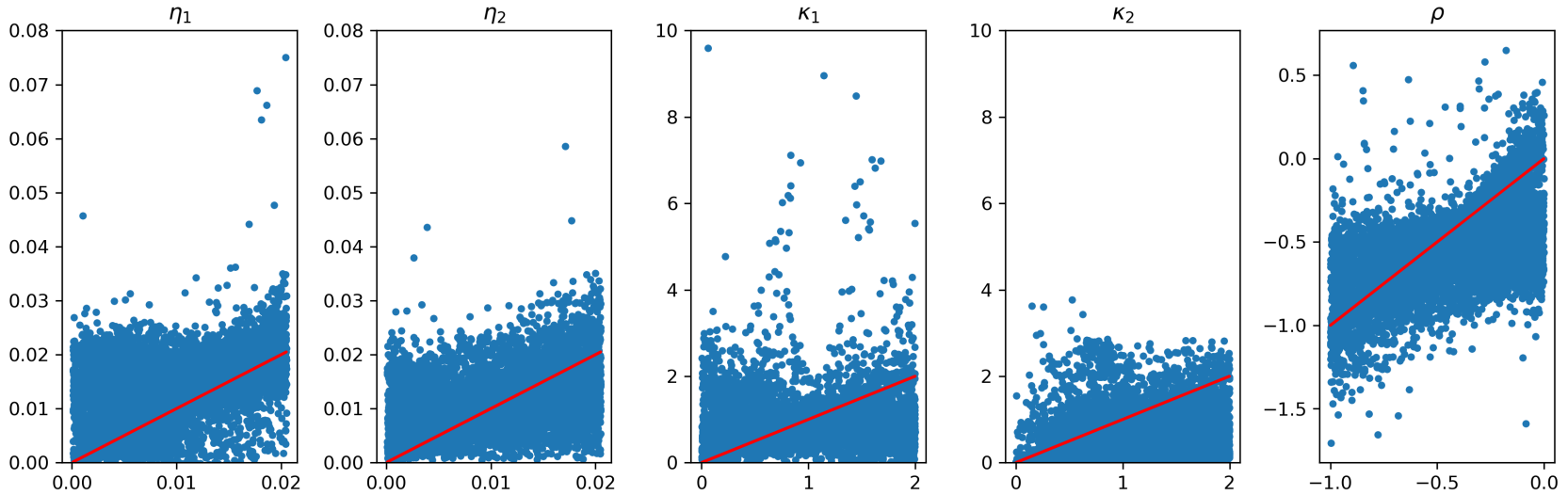


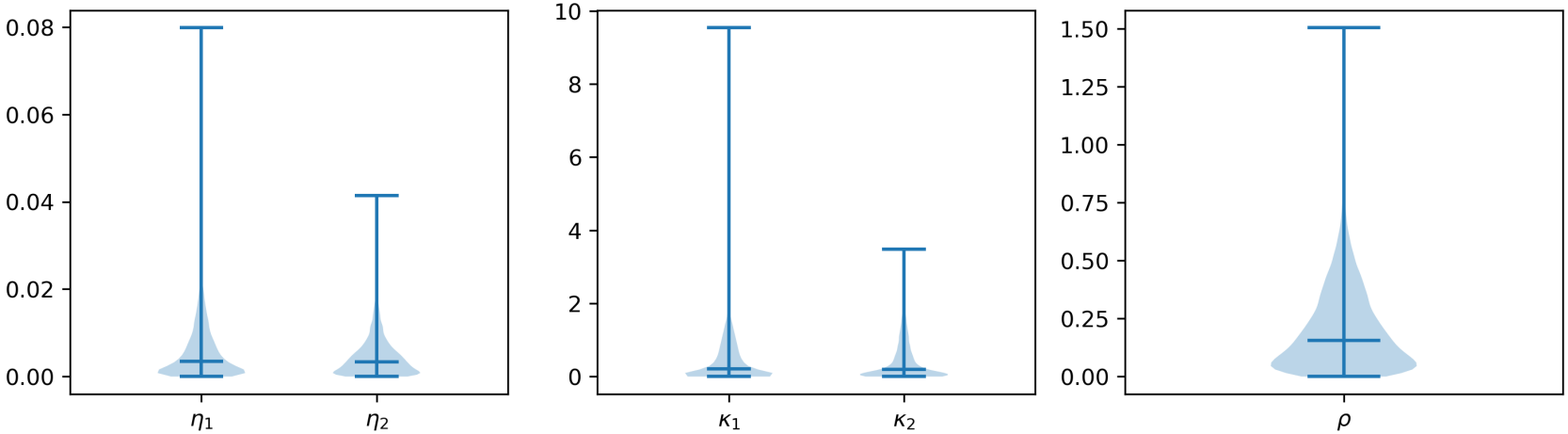
Fehler bei den Testdaten

SciComp

Fehler bei den Parametern

2. Versuch





Geschätzte Parameter

SciComp

Zwischenergebnisse

- Neuronale Netze können Bewertungsmodelle sehr genau approximieren und erlauben eine deutliche Beschleunigung der Berechnungen
- Insbesondere bieten sie eine Ergänzung für Bewertungen, die PDEs oder MC erfordern
- Kalibrierung bleibt Alchemie

- Nächste Schritte
  - Konzentration auf Exposure-Berechnung
  - Neural Structured Learning

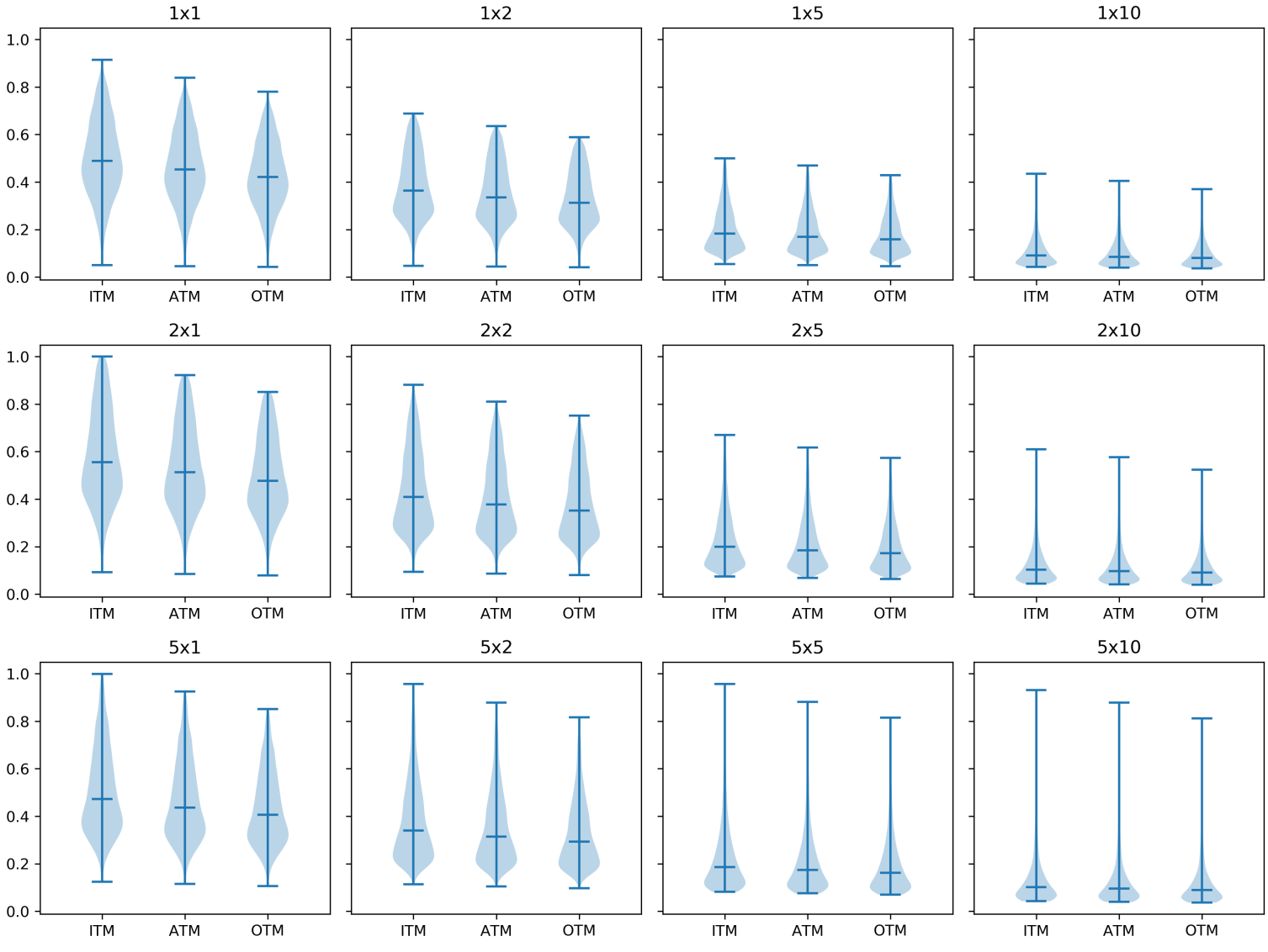
Literatur I

B. Hernandez.
Model calibration with neural networks
Risk Magazin, 2017.

B. Horvath, A. Muguruza, M. Tomas.
Deep Learning Volatility: A deep neural network perspective on pricing and calibration in (rough) volatility models
SSRN-id3322085, 2019.

A. Itkin.
Deep learning calibration of option pricing models: some pitfalls and solutions.
arXiv:1906.03507, 2019

Backup I



Swaptionwerte der Testdaten

SciComp