Calibration of the time-dependent mean reversion parameter in the Hull-White model using neural networks

by

GEORGIOS MOYSIADIS
11126957

36EC
10.2017 - 04.2018

Supervisor: Prof Dr M W van SOMEREN
Assessor: Prof Dr B D KANDHAI

Daily supervisor:
Mr I ANAGNOSTOU MSc
Mr T van der Laan MSc
Abstract

Interest rate models are widely used for simulations of interest rate movements and pricing of interest rate derivatives. In order to calibrate their parameters, several strategies and methods have been proposed. We focus on the Hull-White model, for which we develop a technique for calibrating the speed of mean reversion. This parameter is treated by most existing methods as a constant. We examine the theoretical time-dependent version of mean reversion function and propose a neural network approach to perform the calibration based solely on historical interest rate data. Our results are compared with those obtained by the most widely used methods, linear regression and generic global optimizer. The experiments indicate the suitability of depth-wise convolution over long-short memory modules and prove the advantages of our approach over the existing procedures. We manage to use the knowledge acquired from one market to another, while studying the effects of different subsets of maturities. The proposed models produce mean reversion values that are comparable to rolling-window linear regression’s results, allowing for greater flexibility while being less sensitive to turbulent markets.
Acknowledgements

I would like to thank Ioannis Anagnostou for the mentoring and all the support throughout the duration of this thesis. Despite his busy schedule, he always had time for enlightening conversations that taught me a lot. I would also like to thank Tamis van der Laan for his insightful feedback.

I am grateful to Drona Kandhai and Maarteen van Someren for their guidance and also to Sumit Sourabh, Markus Hofer, Jan Kort and the rest of the Quantitative analytics team of ING Bank for their helpful comments.

My gratitude goes to my people for their resilience and support during this period.
Glossary

**strike (price):** The price at which a specific derivative contract can be exercised. The term is mostly used to describe stock and index options in which strike prices are specified in the contract.

**put/call option:** An option to sell/buy assets at an agreed price on or before a particular date.

**arbitrage-free:** A situation in which all relevant assets are priced appropriately and there is no way for one’s gains to outpace market gains without taking more risk. Assuming an arbitrage-free condition is important in financial models, thought its existence is mainly theoretical.

**spot rate:** The price quoted for immediate agreement on a commodity, a security or a currency.

**forward rate:** A forward rate is an interest rate applicable to a financial transaction that will take place in the future. Forward rates are calculated from the spot rate, and are adjusted for the cost of carry to determine the future interest rate that equates the total return of a longer-term investment with a strategy of rolling over a shorter-term investment.

**swap par rate:** The value of the fixed rate which gives the swap a zero present value or the fixed rate that will make the value of the fixed leg equal to the value of the floating leg.

**time to maturity:** The remaining life of an instrument.

**coupon:** The annual interest rate paid on a bond, expressed as a percentage of the stated value of an issued security.

**zero-coupon bond:** A contract that guarantees its holder the payment of one unit of currency at time T, with no intermediate payments[1].

**compound interest:** Interest calculated on the initial principal and also on the accumulated interest of previous periods of a deposit or loan.

**discount factor:** The factor by which a future cash flow must be multiplied in order to obtain the present value.

**simply-compounded spot interest rate:** The constant rate at which an investment has to be made to produce an amount of one unit of currency at maturity, starting from P(t, T) (price) units of currency at time t, when accruing occurs proportionally to the investment time [1].

**affine function:** An affine function calculates an affine transformation. Affine transformation is a linear mapping method that preserves points, straight lines, and planes. Sets of parallel lines remain parallel after an affine transformation. The general equation for an affine function in 1D is: \( y = Ax + c \).
# Contents

Abstract i

Acknowledgements ii

Glossary iii

1 Introduction 1
   1.1 Machine learning and hand-crafted computational modelling . . . . . . . . . . . . 1
   1.2 Overview and structure . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 2

2 Background 3
   2.1 Neural networks . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3
      2.1.1 Convolution . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 4
      2.1.2 Recurrent . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 5
   2.2 Interest rate and derivatives . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 6
      2.2.1 Term structure of interest rate . . . . . . . . . . . . . . . . . . . . . . . . 7
      2.2.2 Forward rate agreements . . . . . . . . . . . . . . . . . . . . . . . . . . . . 8
      2.2.3 Swaps . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 9
      2.2.4 Swaptions . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 9
   2.3 Hull-White short rate model . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 10
      2.3.1 Mean reversion . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 10
   2.4 Research questions . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 11

3 Related work 12
   3.1 Calibration methods . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 12
      3.1.1 Strategies . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 12
      3.1.2 Calibration techniques . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 13
   3.2 Machine learning for model calibration . . . . . . . . . . . . . . . . . . . . . . . . 16

4 Mean reversion in the Hull-White model 20
   4.1 Hull-White solution . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 20
   4.2 The cyclical theta problem . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 21

5 Calibrating with neural networks 23
   5.1 Studying the data . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 23
   5.2 Neural network with respect to Hull-White’s mean reversion . . . . . . . . . . . . 25
      5.2.1 Can theta be replaced? . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 26
   5.3 Models . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 26
      5.3.1 Convolution architecture . . . . . . . . . . . . . . . . . . . . . . . . . . . . 27
      5.3.2 Recurrent architecture . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 28
   5.4 Mapping . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 28
   5.5 Default calibration with QuantLib and linear regression . . . . . . . . . . . . . . 29
6 Results

6.1 Approximating the default calibration ........................................ 31
6.2 Linear regression ................................................................. 32
6.3 LSTM and CNN ................................................................. 33
6.4 Using forward rate prime ....................................................... 36
6.5 Market knowledge transfer ..................................................... 37
6.6 Isolating initial rate ............................................................. 39
6.7 LM results ............................................................... 40

7 Discussion ................................................................. 42

8 Conclusion and future work ........................................ 47

References ................................................................. 50

Appendix ................................................................. 54
1 Introduction

1.1 Machine learning and hand-crafted computational modelling

In recent years, several practical applications of deep neural networks [2][3] have emerged to provide solutions for a variety of complex problems. Some notable examples are self-driving autonomous cars, medical imaging and speech recognition. The availability of greater data volumes and significantly more powerful computer hardware is enabling deep learning to transform many fields. Finance could not be an exception; there has been a number of attempts to utilize neural networks in order to predict future prices [4], support decision making [5], and construct portfolios [6].

By definition, models are used to represent complex and elaborate phenomena, capturing only some essential facets of reality. Hand-crafted models are usually more accessible than the actual subject of study, and are convenient for practitioners because of their explainability. Yet, the main intellectual concern is whether all the crucial characteristics of the problem are sufficiently embodied. The purpose and the use of a model can define the criteria to choose the most appropriate, weighing the importance of simplification and explainability.

The range and quantities of the data that is currently being collected, exceed the capabilities of any human to analyze. Existing statistical techniques suffer and hand-crafted models often prove to be insufficient to handle the sizes and complexity of the datasets. However, paradigms of machine learning, such as neural networks, address these shortcomings. They are able to identify features that are used to connect seemingly unrelated inputs, while capturing complex relations within large datasets. This is achieved without explicit parameterization, which allows flexibility in the design of a neural model, but can decrease our ability to explain its underlying decisions.

The structure of a neural network offers distinct properties that affects its performance. Like most computational models, they are defined based on the type of the problem and the form of the underlying data. The analysis of the data contributes to the recognition of special circumstances that can determine the introduction of specific architectural features. Essentially, the findings of such analysis are used to specify the desired characteristics of the network modules to be used, outlining the construction of a suitable architecture.

Adapting a data-driven approach, such as neural networks, inevitably leads to partly disregarding existing models that capture our current comprehension of a problem. Previous rigorous studies, well-established knowledge and experience are incorporated in mathematical expressions that define computational models. They encode deeper insight based on expertise, data and wider view over certain aspects of the problem, that are usually not utilized in machine learning approaches.

The advantage of such handcrafted methods, compared to neural networks, lies on the ability of the expert author to understand abstract concepts, that cannot be sufficiently expressed in existing metrics. In other words, the capability to identify forces and behaviors that continuously affect the movement of a value that cannot be directly quantified. For example, some recurring political decisions that drive the evolution of a stock. The effects of such decisions may be implicitly embodied in certain measures, but do not describe the complete causal relation. Moreover, the patterns that are visible, in terms of data, can occur in intervals that may be difficult to be handled by any data-driven approach.

One such example is the cyclical movement of interest rates that is caused by economic
and political factors. Methods that rely on month-long data input may not be able to identify the mean-reverting behavior, since this phenomenon is more pronounced over longer periods. The expert approach to describe this behavior in mathematical terms, is the introduction of a parameter that is not directly observed in the data, but depends on measurable variables. This quantifiable metric, the speed of mean-reversion, contributes to the simulation of the evolution of interest rates, as a factor pushing towards a long term average. A variety of interest rate models adopted this parameter, offering elaborate definitions, but preserving the theoretical interpretation. Calculating it, requires the extraction of patterns that are described in historical interest rate data, but also the recognition of more complex relations between market segments.

Existing methodologies, such as linear regression, succeed to explain simple linear relations in market data. Neural networks have been proposed for the calibration of the speed of mean-reversion, as they are able to find and learn more complicated structures and associations whose existence is apparent. Using neural networks to estimate variables for explicit computational models enables the experimentation with more complex and larger datasets, which ultimately can improve their performance.

1.2 Overview and structure

This thesis is structured as follows. In section 2, we provide an introduction to neural networks, interest rate models and present the basics of the Hull-White model, which is going to be our main focus. Upon setting the challenges we meet, we formulate the research questions. In section 3 we synthesize the literature related to the specific dynamics of IRMs focusing on approaches involving neural networks and machine learning techniques in some way. In section 4 we elaborate on Hull-White dynamics, we study the calibration process and provide the theoretical basis for our approach. In section 5 we discuss our methodology, data pre-processing and the structure of the neural networks. In section 6 we address our research questions through empirical experiments and present our results. Next in section 7 we discuss our findings and, finally, in section 8 we conclude and suggest directions for future research.
2 Background

2.1 Neural networks

Originally introduced as a concept by Warren Sturgis McCulloch [7] in 1943, artificial neural networks developed to a fast growing trend in machine learning. The simplest of them, the perceptron, shares similarities with regressions, but their evolution led to the emergence of much more complex paradigms. They are devised as a computational equivalent of the human brain; every neuron is a computation node that applies a non-linear function \( f \) (e.g. sigmoid), which is activated depending on the input. These nodes, like in the human brain, are interconnected. They form layers that move information forward from one layer of neurons to the next. The neurons of each layer are not connected, allowing communication only with previous and succeeding layers.

The information flow, considering the supervised learning paradigm, starts by the network’s input \( x \), which is transformed from layer to layer resulting a value \( Y \) that should match a predefined outcome. The learning capacity of the network relies on the weights \( w \), that connect the computation nodes, and are trained based on the error calculated comparing the network’s output to the expected result. This error is then back-propagated to alter the value of the weights. The full network expresses a fully differentiable function that is described and learned by the training data. In that sense, a neural network is a generic function approximator.

More explicitly, consider a dataset which consists of the \( x_1, x_2 \) data, e.g. the average price and its variance for a day, as the input, and \( Y \) the close price of the next day, as the output. This is the value that the neural network will learn to predict. The training of the model is the procedure of adjusting the weights \( w \) connecting the neurons of the model. This is achieved by minimizing a cost function of which the simplest form can be \( C = \sum(Y - \hat{Y}) \). The cost function, as the name suggests, is the cost of making a prediction using the neural network. It is a measure of accuracy of the predicted value, \( \hat{Y} \), with respect to the observed value, \( Y \). Various types of cost functions are used in practice, depending on the formulation of the problem.

The neural network is trained by computing the cost function for the input data given a set of weights. Then the training algorithm moves backwards and adjusts the weights calculating the partial derivatives of the node’s activation function. These steps are repeated until certain
conditions are met, regarding the minimization of the cost function. The process of applying the errors to adjust the weights, termed backpropagation, is continuously advancing, significantly improving the learning performance and enabling even larger datasets to be handled.

Through the evolution of neural networks, more complex computation modules were developed to work along simple activation nodes. Deeper architectures with many layers were proposed and studied for their performance with a variety of data. Currently, deep neural networks are increasingly used for image processing, reconstruction and identification. Generally, less deep networks related to time-series find application with complex language processing and temporal pattern identification. Several modules are built for these purposes, two basic of which are convolution and recurrent modules.

Fig. 2.2: Simple neural network

Convolution modules are the basis of very popular image recognition networks [8] [3] but also used with financial time-series for forecasting and other purposes [9]. Recurrent networks, and specifically LSTM modules, are mostly used with time-series or with data that are ruled by complex time dependencies. Both modules are used similar to simple neural network nodes, both have trainable weights and receive the same input, what differs is the way this input is processed.

2.1.1 Convolution

Convolutional neural networks (CNN) can be seen as the organizational analogy of animal visual cortex, where a neuron responds to stimulus only in a region of the visual field, while the regions of different neurons partially overlap. In practice, the convolution modules were proposed to address limitations of hand-crafted feature extractors for images, such as the need for pre-processing so that the input data would meet certain assumptions [10]. This mechanism is realized with neural networks by applying, along the input data, a convolution filter. This is limited in size, smaller than the image or time-series.

Simple network nodes that are fully connected similar to figure 2.2, consist the most generic module that can be used. Theoretically, a fully connected network is able, under certain conditions, to approximate an arbitrary function[11] and learn all complex non-linear relations that can be learned by any neural model [10]. In practice, they do not scale well for high dimen-
sional data, since they require increasingly large training sets proportional to the number of the weights. This issue becomes worse when the problem to be addressed exhibits translation or scaling invariance, e.g. object recognition. This issue is solved by convolutional networks that typically require significantly fewer weights to be trained and scale better.

![Convolution operation in 1D space](image)

Convolution is a mathematical operation that combines two functions. The integral of their pointwise multiplication is calculated, to result in a third function that can be seen as a modified version of the original two. In the neural network module, the aforementioned filter, or kernel, is made up by trainable weights applied in this way on the segment of data (fig 2.3) that lies in their receptive field. The output of this operation typically undergoes pooling, reducing the number of features, and depending on the architecture, the result may be fed to a fully connected layer of simple nodes. For images, same sized filters are applied in partially overlapping windows. In time-series the specified window will move over the signal with fixed step size. The size of the window is a hyper-parameter and correctly defining it leads to significant gains in performance.

### 2.1.2 Recurrent

Both fully connected and convolutional networks follow the same pattern with respect to the data flow, recurrent networks (RNN) introduced a different approach. Instead of only allowing information move forward from previous layer to the next, RNNs connect the output of a layer back to its input with the appropriate trainable weights, while feeding the results to the next layer as well. Together with this idea, several concepts were created; the internal module state (memory), time-varying activation and backpropagation through time. The RNN family counts many individual modules that rely on different assumptions but all attempt to learn temporal dependencies of some form.

The most notable module of the RNN family, long-short term memory (LSTM [12]), was developed to address several inefficiencies of simple recurrent networks and be capable of learning both short and longer time-dependencies. Network architectures consisted only by LSTM modules have been applied successfully in diverse areas and different problems; speech recognition [13], translation [14], but in combination with other models most notably CNNs, for image content explaining [15].
In figure 2.4 we see the LSTM module, termed cell. The cell state $C_t$, or cell’s memory, carries information from one time step to the next. It is altered and regulated by gates that apply their output either by multiplying $\times$ or adding $+$. The leftmost $\sigma$, which represents the sigmoid function with output in the range $[0,1]$, specifies how much information should be kept in the cell state from the previous step. It combines the output of the previous time-step, $h_{t-1}$ and current input $X_t$, to determine how much each number in the cell memory is needed, 0 to forget it, 1 to keep it.

Consider a model that learns to predict the next word of a sentence based on the previous ones. In this case, the cell state may include whether the current subject is in plural or singular, so that the correct form of the verb can be used. When a new subject is seen this information should be forgotten. Similarly, if the model is trying to predict regime changes in interest rate market, the current cell state may hold information about the relative position of the highest maturities. When a new regime is seen this information should be forgotten.

The next step, which is the second sigmoid and $\tanh$ activation, specifies how much of the new information should be added to the cell state. The sigmoid decides how much and which values will be updated and the $\tanh$ creates the new cell state candidates. This information is combined by the multiply ($\times$) operation and then added ($+$) on the current cell state. This, in combination with the previous step, concludes the update of current cell state.

The final step produces the output of the cell by first deciding how to filter our current state (right-most sigmoid) and applies this filter to the updated cell state after it is re-scaled by $\tanh$ to produce $h_t$.

### 2.2 Interest rate and derivatives

The interest rate has several functions in an economy. It is one of the main tools of monetary policy for the governments to steer economic variables and affect factors of social significance, such as unemployment, inflation and investment. Interest rate determines the conditions that goods of today will be traded in the future. Almost all banking activities, involving lending of capital, are influenced by its level and the expectations for its movement, since it sets the margin for profitability.

To provide a more intuitive view on the significance of interest rate, consider that central
banks hold money for commercial banks as a reserve. The level of interest rate can affect the level at which this money is distributed (lent or invested). For example, if the central bank’s policy seeks for an increase in spending and investment to stimulate economic activity, the interest rate will be lowered making the reserve less profitable, pushing commercial banks to invest in more profitable positions. If interest rate becomes negative then the incentives are even stronger, since commercial banks are charged interest on the reserve. Central bank’s interest rates are mostly positively correlated to the bank’s offered interest rates. In a similar fashion, low offered rates will make saving for households less profitable, encouraging them to spend now instead of saving.

On the other end, scaling up interest rates can be used to control inflation. This will increase the borrowing costs, resulting to the decline of borrowing from business and individuals. In turn, the spending level will be reduced, while the demand of goods and services will drop together with inflation. Many more financial decisions involve this trade-off between present and future consumption, interest rate is a crucial variable in this choice.

However, central bank interest rates (government rates) are only one reference point for financial transactions; the price at which financial institutions are borrowing and lending money to each other is also a widely used index (interbank rate). Libor (London interbank offered rate) is considered the most important interbank rate used for contracts. It will be the index of our experiments throughout this project.

Interest rates are used as a reference point not only for lending but also for the derivative market. Derivatives that rely on interest rates have become an exponentially growing market over the years. This has led to an increase of connectedness between institutions in the economy but also the sharing of risk. In normal lending, one counterparty is seen as risk free party that is exposed to credit risk by lending capital. For example, a bank offering a mortgage loan to an individual. In the derivative market, the traded instruments, and in particular interest rate swaps, expose both parties to the risk of loss. In this way, interest rates constitute a risk factor for counterparties exposed to such products.

Interest rate models, in the context of derivatives, arose from the need to model the future evolution of interest rate. They are used to estimate counterparty risk, simulate future scenarios, secure arbitrage-free conditions but also for the needs of valuing instruments such as bonds.

The particular IRM that we are going to study fall into affine term structure model (ATSM) category, as its basic assumption is that the unobservable short rate that it attempts to model, is an affine function of some latent factors. Short-rate is the interest rate at which money is borrowed for a period of time where \( \Delta t \to 0 \). These type of models incorporate stochastic processes to simulate the movement of the yield curve, combined with free parameters that are calibrated based on historical market data, adding flexibility and allowing them to be used for different products and market conditions.

2.2.1 Term structure of interest rate

The term structure of interest rate or yield curve, also referred as zero-coupon curve, is made up of interest rates paid by zero-coupon bonds of different maturities but of the same level of risk. It is the depiction of the function which maps maturities (in time units) into rates. The yield curve can be viewed as an indicator of the current market expectations for the future movement of interest rates.
A yield curve, such as fig. 2.5, shows that the short term yields are lower than the long term yields, so the curve slopes upward, recording the expectation that the short term rates are going to increase. The other two distinctive shapes are the flat and the inverted curve. The flat yield curve indicates that investors are expecting the interest rate to remain on the same level as today. Whereas the inverted curve, sloping downward, indicates that the market is expecting the short term interest rates to drop.

However, this interpretation is not universally accepted. Alternative interpretations for the slope of the yield curve suggest that the curve should be treated in segments. Each segment is populated by investors with a particular preference for investing to assets with maturities within this market. Thus, the rates are determined only by supply and demand within each segment. Both theories have extensions that contribute to the explanation of the dynamics of the yield curve, while the more widely accepted one tends to be the that of market expectations[18].

The yield curve is usually calculated on selected or all bonds traded, curve 2.5 is based on AAA-rated bonds only. IRMs rely on the current term structure to compute forward rates but also as the input of past (t-x) rates. In the following sections we will elaborate further.

### 2.2.2 Forward rate agreements

A forward rate agreement (FRA) is a contract between two counterparties that determines the rate of interest to be paid or received by the contract holder. This contract obligates the payer to pay a fixed rate $K$ and the receiver to pay a floating rate from future expiry date $T$ until maturity $S$. The floating rate is generally a less predictable reference e.g. libor spot rate $L$. The rates are applied on the nominal value $N$ of the contract and involves an one-off transaction conducted in the beginning of the of the forward period $T$. Following [1] p.11-13 notation the value of the contract at maturity can be written:

$$N\tau(T, S)(K - L(T, S))$$

(2.1)
where $\tau(t, T)$ refers to the time distance $T - t$, and $L(T, S)$ to the simply-compounded spot interest rate from expiry to maturity which can be expressed:

$$L(t, T) = \frac{1 - P(t, T)}{\tau(t, T)P(t, T)} \tag{2.2}$$

where $P(t, T)$ denotes the price at time $T$. We refer to time difference as $\tau$ function to preserve generality, since the measure of time varies depending on the market. The simple subtraction makes sense when we deal with real numbers, but markets use a variety of day-count conventions to define the amount of time between two dates. For our purposes we use Actual/360 convention since it is used for Libor and Actual/365 when we deal with GBP[19]. The full list of day-count conventions can be found in [20].

In order to get the total value of the contract, expression (2.1) is multiplied with the price function $P(t, S)$, resulting to the total value of the FRA at time $t$:

$$FRA(t, T, S, \tau(T, S), N, K) = N \left( P(t, S)\tau(T, S)K - \frac{P(t, S)}{P(T, S)} + P(t, S) \right) \tag{2.3}$$

2.2.3 Swaps

Interest rate swaps are the generalization of forward rate agreements. Instead of one cash flow, the contract obligates the two counterparties to exchange payments starting from a future time at pre-specified dates. The two parties agree to exchange cash flows, one pay a fixed rate and the other a floating rate, termed as the two legs of the swap. The floating rate is the value of the reference index (e.g. Libor) at the moments of transactions, which are conducted based on a notional principal at the end of each period, in contrast to FRAs. In general, there is no need for the two payments to take place in the same day or under the same day-count convention [21]. It is actually very common for the two legs to follow different day-count conventions [1]. Moreover, swap contracts can be used with different currencies and are not limited to only interest rate indices as reference. For example, in an equity swap, Libor can be used as reference point for one leg and a pre-agreed rate upon the index of stocks relative to the notional amount of the contract for the other leg.

2.2.4 Swaptions

A swaption is a “composite” type of derivative, in the sense that it combines two kinds of contracts, options and swaps. Simply put, it is an option to enter an interest rate swap. These contracts give the holder the right but not the obligation, to enter a swap with pre-agreed terms within a period of time (tenor). There are three main categories of swaptions, Bermudan, European and American with the difference found at the time-points, during the life of the contract, at which the holder can activate the swap agreement.

Swaptions are distinguished by the counterparty that receives the fixed leg, termed payer and receiver swaptions. The counterparty holding a payer swaption has the option to enter a swap in which it will be paying the fixed leg of the swap and will be receiving the floating leg. Likewise, the counterparty holding of the a receiver swaption has the option to enter a swap paying the floating leg and receiving the fixed leg. The same naming distinction holds for swaps as well. Interest rate swaptions are quoted in terms of the volatilities of the forward swap or Libor rates which are their reference points.
The difficulty of valuing either swaps or swaptions originates in the general inability to accurately and reliably predict the future movement of interest rate. It is not described by a deterministic function but is speculated by the shape of the yield curve. The dynamics of interest rate have been studied extensively leading to the inception of a family of interest rate models that aim to predict the future movement of the rate. They rely on one or more stochastic terms and incorporate assumptions of temporal relations.

2.3 Hull-White short rate model

The models we consider, describe interest rate movements driven by only one source of risk, one source of uncertainty, hence one-factor model. This translates in mathematical terms having only one factor driven by a stochastic process. Apart from the stochastic term, the models are defined under the assumption that the future interest rate is a function of the current rates and that their movement is mean reverting. We will elaborate in mean reversion in the next sections. The first model to introduce the mean reverting behaviour of interest rate was proposed by Vasicek [22]. The Hull-White [23] model is considered its extension. The Hull-White SDE reads:

\[
dr(t) = (\theta(t) - \alpha r(t))dt + \sigma(t)dW(t)
\] (2.4)

where \(\theta\) stands for the long-term mean, \(\alpha\) the mean reversion, \(\sigma\) the volatility parameter and \(W\) the stochastic factor, a Wiener process. Calibrating the model refers to the process of determining the parameters \(\alpha\) and \(\sigma\) based on historical data. \(\theta(t)\) is generally selected so that the model fits the initial term structure using the instantaneous forward rate. However, its calculation involves both \(\sigma\) and \(\alpha\), increasing the complexity when both are time-dependent functions.

Studying the equation (2.4) we observe the aforementioned dependence on previous instances of interest rate which become even more obvious if we consider that \(\theta\) indirectly relies on the term structure of interest rate as well. Clearly this model incorporates both temporal patterns, expressed as temporal dependencies and the market’s current expectations, while the mean reversion term suggests a cyclic behaviour, also observed in many other financial indicators.

2.3.1 Mean reversion

The concept of mean reversion suggests that the interest rates cannot increase indefinitely, like stocks, but they tend to move towards a mean [24], as they are limited by economic and political factors. There is more than one definition of mean reversion varying not only by model, but from market to market as well. Mean reversion can be defined by historical floors and peaks or by the autocorrelation of the average return of an asset [25]. In the Vasicek model family, it is defined against the long term mean value towards which the rate is moving with a certain speed.

The performance of Hull-White model is significantly affected by the level of mean reversion. A small value would produce more trending simulation paths, while a larger value can result in steady evolution of interest rate. A mean reversion that does not reflect the actual situation can lead to miss-calculation of risk and exposure which in turn may result in non-optimal use of capital.

The time-dependent nature of long term mean springs from the need of interest rate models to fit to the initial term structure. By allowing the mean reversion to be a function of time as
well, the ability of the model to provide good fit to a continuously changing term structure is increased, strengthening the theoretical soundness. Correctly calculating this function secures the trustworthiness of simulations and the more precise estimation of the evolution of interest rate.

2.4 Research questions

From a data-science perspective we could easily identify by visual inspection and some statistical analysis that interest rates are following some form of temporal (cyclical) patterns. However, a rigorous understanding of the mean reversion dynamics requires deeper knowledge and analysis of the market. Exploiting the empirical knowledge which we have summarized in the previous section, we seek to synthesize a method that can effectively address the following questions.

**Research question 1**

By learning temporal patterns from financial data that incorporate the movement of interest rates, can we recreate the dynamics described by the Hull-White model parameters in order to build a fast and effective calibration algorithm for the speed of mean reversion?

Since Hull-White and previous models are based on mean reversion and long-term mean assumptions that indicate the existence of temporal patterns, there should be evidence to confirm this in historical interest rate data. Driven by this, we suggest to create deep learning models (CNN or LSTM based) that can adequately harness this knowledge from data, to learn temporal dependencies and enable us to enhance the effectiveness of the model.

**Research question 2**

Can we map high dimensional financial data to simpler structures in order to exploit complex dependencies? Can we take advantage of multiple product families or different markets to learn common behaviours?

Successfully addressing the first part of this question will allow us to decorrelate financial factors, such different maturity segments of yield curve, and maximize the informational gain from the underlying market indicators. Doing that, should enable us to combat overfitting by using information from other markets or products without reusing common features as independent and open the way for a general non-product based calibration procedure.
3 Related work

The procedure that determines the two parameters of Hull-White model, speed of mean reversion and volatility, is a topic that has been studied by both academics and practitioners, offering a variety of solutions that not always yield similar results. Each method has its restrictions and limitations, while the evaluation is generally conducted based on the quality of fit of the calibrated model to historical prices. The particular result of the calibration is difficult to be compared with regard to the theoretical meaning of the corresponding parameter. Mean reversion, theoretically expresses the speed of the movement towards a long term mean. However, the literature suggests that being consistent with this definition is not always the goal of the calibration process.

3.1 Calibration methods

In this section, along studying previous work, we will attempt to provide an intuitive explanation of the dynamics of the two parameters, as well as the particular challenges that the studied techniques have met. Then we will be able to swiftly move to our neural network approach. The reader should keep in mind that calibration is seen as a reverse engineering process where the model parameters are reconstructed from market prices.

Before proceeding, it is required to outline the basic properties and the form of the data used in these procedures. The first input to our model is the term structure of interest rate, which is also used for the calculation of the long term mean. Since we need to work on continuous time but the yield curve is made up of a limited number of values, it is common practice to interpolate these values to get a continuous approximation of the slope. The interpolation method varies depending on the approach. In our study the cubic spline interpolation is used similar to [26][27][28].

The second input to the model is the instrument’s volatility, the parameter that describes the uncertainty of the price movement of the underlying asset. In order to acquire the volatilities, in general, the reverse Black-Scholes [29] formula is used, taking as input the market prices to determine the theoretical volatility value, termed implied volatility. For swaptions, this metric forms the volatility surface which maps each instrument’s tenor and maturity to the respective volatility value. Implied volatilities could express the market’s expectations about future volatility in the forward rates over the life of the option and are indicators of the future degree of uncertainty.

3.1.1 Strategies

Starting with the strategy of calibration, in [17] three topics are explored:

- Whether the parameters should be constant or time-dependent
- The specification of the products to calibrate on
- Whether the two parameters should be calibrated together or separately

The first two topics refer to decisions that have to be made, and are irrelevant to the actual calibration method to be used. The third topic is closer to that since the value of one parameter affects the levels of the other, so the sequence does affect the overall result.
Deciding which parameters should be considered constant, in reality, is changing the whole model approach, since it alters the fundamental relations between the parameters. Consider that for the Hull-White model, \( \theta \) relies on the value of mean reversion; setting the latter a constant by hand may simplify the model making it easier to handle, but it changes the dynamics of the long-term mean. The degree that this influences the overall result is studied by a series of tests in [17].

The authors conduct multiple experiments in order to determine how calibration is affected by different product maturities. Specifically, they study the effects of co-terminal calibration i.e. calibrating on one or two swaptions per maturity. The outcome of their tests indicate that the partial use of the volatility slope result in a volatile movement of mean reversion, which, as they underline, is not suitable for pricing.

Another finding worth noting is that the implied volatility curve slopes downward as the mean reversion parameter increases. On the other end, changing the volatility parameter by hand does not have the same effect on the mean reversion which vary insignificantly. This offers a view on the dynamics of the two parameters, that can prove to be valuable practitioners. The main conclusion of this practical approach suggests that the calibration algorithms may provide more intuitive results when treated separately, preserving the fidelity of the calibrated variables to their theoretical definition. Calculating both values in parallel, aiming for the best model fit, can lead to systematic deviation, making the model biased to the trends learned from the calibration data.

### 3.1.2 Calibration techniques

The actual parameter calibration is approached by several methodologies with different models and assumptions. The common base of the existing methodologies, is that the parameters should enable the Hull-White model to fit the observed market values. This is achieved by either fitting the market data directly with certain simplifications that allow fast calibration, or follow the model assumptions and approximate the implied value. We also visit more elaborate and restrictive methods, that can be applied under certain conditions and for specific instruments.

#### Linear regression

Consider the Hull-White model with constant \( \theta, \alpha \) and \( \sigma \)

\[
\frac{dr(t)}{dt} = (\theta - \alpha r(t)) dt + \sigma dW(t)
\]

Using Ito’s lemma it is proved:

\[
 r(t) = r(s)e^{-\alpha(t-s)} + \frac{\theta}{\alpha} \left(1 - e^{-\alpha(t-s)}\right) + \sigma e^{-\alpha(t-s)} \int_s^t e^{\alpha(t-u)} dW(u) \tag{3.2}
\]

which follows the distribution:

\[
r(t) \sim \mathcal{N} \left( r(s)e^{-\alpha(t-s)} + \frac{\theta}{\alpha} \left(1 - e^{-\alpha(t-s)}\right), \frac{\sigma^2}{2\alpha} \left(1 - e^{-2\alpha(t-s)}\right) \right)
\]

where \( s < t \).

Consider \( s = t - 1 \), the previous time point of the state \( t \). The relation between \( r(t) \) and
\( r(s) \) is linear and can be written as:

\[
r(t) = \hat{\alpha}r(s) + \hat{\theta} + \epsilon(t)
\]  

(3.3)

Based on this observation, a widely used method to calibrate Hull-White is to fit a linear model by minimizing the squared error [28] and use the trained parameters to calculate the model’s values. The output of the regressions is mapped back in terms of Hull-White as follows:

\[
\hat{\alpha} = e^{-\alpha(t-s)} \Rightarrow \alpha = -\frac{\ln \hat{\alpha}}{t-s}
\]

\[
\hat{\theta} = \frac{\theta}{\alpha} \left(1 - e^{-\alpha(t-s)}\right) \Rightarrow \theta = \frac{\alpha \hat{\theta}}{1 - e^{-\alpha(t-s)}}
\]

\[
sd_{\epsilon} = \sigma \sqrt{\frac{1 - e^{-2\alpha(t-s)}}{2\alpha}} \Rightarrow \sigma = sd_{\epsilon} \sqrt{\frac{-2 \ln \hat{\alpha}}{(1 - \hat{\alpha}^2)(t-s)}}
\]  

(3.4)

Linear regression is trained on historical market data, the length of which is a choice of the user and can be determined by empirical knowledge. Our suggestion on this matter is based on the hypothesis that if there is a great change in the market values, i.e. a highly abnormal period, this portion of historical data does not reflect the current condition of the market. That means that it is probably not suitable to train a linear model on, when the market turbulence has passed, since the relation between consecutive points was affected by forces that do not exist in the market after this period.

**Levenberg-Marquardt and simulated annealing**

In order to evaluate the quality of Hull-White parameter fit, one can calculate the swaption price that the model yields, the implied price \( P^\text{mod}_i \), and compare it with the observed market price \( P^\text{market}_i \). The objective of the optimization procedure is defined:

\[
\min_v \sum w_i \left( P^\text{market}_i - P^\text{mod}_i \right)^2
\]

(3.5)

where \( v \) is the vector of parameter to be optimized, which corresponds to the input parameters.

Levenberg-Marquardt [30] and simulated annealing [31] are very well studied algorithms that are, generally, able to find global optimum when local optima are present as well. To the extent we are concerned, both algorithms result in a very good fit and although we use them as a reference point for our results, the comparison or a deeper study of the two exceeds the scope of this work.

**Jamshidian decomposition**

In [32] Farshid Jamshidian made a simple but very useful observation which developed into a mathematical trick: Consider a sequence of monotonic increasing functions of a real variable such \( f_i(x) \in [0, \infty) \), a random variable \( W \) and a constant \( K \geq 0 \). Since \( \sum_i f_i \in [0, \infty) \) is also increasing then there is a unique solution \( W \in \mathcal{R} \) to \( \sum_i f_i(w) = K \)
Since the functions \( f_i \) are increasing we have:

\[
\left( \sum_i f_i(W) - K \right)_+ = \left( \sum_i (f_i(W) - f_i(w)) \right)_+ = \sum_i (f_i(W) - f_i(w))_+ \tag{3.6}
\]

We can see \( f_i(W) \) as the value of an asset and \( K \) the strike price of the option on the portfolio of assets. So we express the price of the option on the portfolio in terms of a portfolio of options on the individual assets \( f_i(W) \) with strike price \( f_i(w) \).

Let’s consider a European style option, i.e. an option that can be exercised at its maturity, with strike price \( X \) and maturity \( T \) on a bond that pays \( n \) coupons \( c_{1\ldots n} \) at times \( T_{1\ldots n} \). Following [1] notation, the coupon baring option price at \( t < T \) is

\[
CBO(t, T, T, c, X) = \sum_{i=1}^{n} c_i ZBO(t, T, T_{i}, X_{i}) \tag{3.7}
\]

where \( ZBO \) denotes the price of European option on zero-coupon bond. In terms of swaptions, the Jamshidian trick translates into that one can calculate the value of a swaption as a combination of put or call options on zero-coupon bonds. Consequently, since the European swaption can be viewed as an option on coupon bearing bond, the swaption price at \( t \) with nominal value \( N \) can be written:

\[
S(t, T, T, c, X) = N \sum_{i=1}^{n} c_i ZBO(t, T, T_{i}, X_{i}) \tag{3.8}
\]

For formal proof of the above we refer to [1] p.75-78 & 112. In terms of Hull-White calibration, Jamshidian decomposition is used to calibrate the volatility of the swaption price.

**SMM approximation**

In contrast to Jamshidian decomposition which is an exact solution in [33] and [34] are proposed two similar models, swap market model (SMM) and libor market model (LMM). Both leading to Black’s formula [35]. The difference between the two models lies on the choice of market rates that follow log-normal processes. In the case of LMM, this is a set of forward Libor rates, for the SMM this is a set of forward swap rates.

The SMM SDE reads:

\[
dS_{n,N-n}(t) = S_{n,N-n}(t)(\mu_{n,N-n}(t)dt + \gamma_{n,N-n}(t)^{\top}dW_t), \quad n = 1,\ldots,N - 1
\]

\( S \) denotes the forward swap rates, that are following log-normal distribution, \( \mu \) the drift function and \( \gamma \) denotes N-1 deterministic Black-type functions to calculate the implied volatility.

Notice that SMM model needs calibration as well, which is done by fitting historical swap rates to the model, similar to Hull-White using a generic optimizer. However, only the observed market prices (swap rates) are needed, contrary to Hull-White, where the unobserved instantaneous forward rates are also needed. For that, SMM can be used to calibrate mean reversion on swap rates and then use it with Hull-White.
3.2 Machine learning for model calibration

Although machine learning and neural networks have been widely used in finance for stock prediction [36][37][9][4], volatility modelling [38], currency exchange rate movement [39] and many more, only a few attempts have been made to address the calibration problem, and specifically, mean reversion estimation. Here we are going to review some of them and try to outline features that could be used in Hull-White calibration.

Consistency hints

In [40] the author studies interest rate models and uses the Vasicek, in particular, for his application. He introduces an enriched expectation maximization (EM) algorithm [41] to force the parameters to be normally distributed, as the assumptions the model dictate. There are a few points that we need to underline from this work; first, the author treats the problem of calibrating a multi-factor model, i.e. the Vasicek model that is applied in a multitude of correlated instruments, which can be translated in using several swap tenors independently. Second, he points out the distinction between a good model fit and a correct parameter fit. In other words, as explained, the values of the parameters that yield acceptable or even the best results with respect to the model output, do not guarantee that they actually reflect the truth with respect to their theoretical definition. This phenomenon inevitably lessens the models’ predictive power by overfitting. In this regard, the term “overfitting” adopts a somewhat wider interpretation of that commonly implied in a machine learning context. Here the model does not only overfits to the underlying data, but it also shifts its theoretical meaning to match certain product prices.

What are the assumptions that are violated by calibrating simultaneously the two parameters $\alpha$ and $\sigma$? The Vasicek model SDE, similar to Hull-White, is expressed as follows:

$$dr(t) = \alpha(\theta - r(t))dt + \sigma dW(t)$$ (3.9)

with the multi factor discretized expression being:

$$\Delta r_n(t) = \alpha_n(\theta_n - r_n(t))\Delta t + \sigma_n w_n \sqrt{\Delta t}$$ (3.10)

where $n$ is the pointer to the particular underlying rate. The author finds that the model assumption on the stochastic term is violated; he shows that by solving for the implied $w_n$, the resulting values violate the statistical property that the time-steps should be normally distributed.

In order to enforce this rule in the calibration procedure, he introduces three error functions that are applied by the EM algorithm. The first, of course, expresses the need that the model should be approximating the real historical price values. The second determines and penalizes the outcome based on the Kullback–Leibler distance, which realizes the enforcement of the normal distribution assumption. The third, enforces the distribution of the initial state. These three error functions apply the “hints” during the calibration to achieve a correct parameter fit. The author claims that this procedure indeed results to a very close fit of the volatility parameter to the actual values, in contrast to the normal calibration. Moreover, it produces a correct statistical interval for model simulations, while the model errors are generally very close to that of the normal calibration.
Nonlinear mean reversion modelling

In [42] the authors propose the use of neural networks to model the non-linear mean reversion and apply their technique calculate the reversion of price discrepancies between a stock and its cross-listed equivalent, i.e. the stocks of the same company traded in different markets and currencies. They explain that this value follows a mean reverting movement; a non-mean reverting behaviour would suggest that the market prices of the same company are not co-integrated. This would open arbitrage opportunities, in reality, it is expected for some such opportunities to appear in high-frequency transactions, but these are difficult to take advantage of and, in practice, are not profitable.

The authors define their feed-forward neural network that is trained to predict the slope, i.e. the average difference between discrepancies $d$ of $m$ consecutive time points:

$$\text{slope}(i) = \frac{1}{m} \sum_{k=i}^{i+m} (d(k + 1) - d(k))$$

We need to point out two important properties of this definition; first, the degree of moving towards a mean value is expressed and evaluated at every time point from real data, i.e. the mean is known. It is not an approximation based on the current expectations, i.e. long-term mean of Hull-White. Second, their formulation provide a freedom over the degree of smoothness: by increasing the simulated steps $m$ they increase the factor that smooths the slope, a property which, as we mentioned earlier, is significant for real world applications.

In the context of IRMs, abrupt changes of mean reversion are probable, but problematic in use. The mean to which the rate is reverting is defined by the current interest rate level, acting in that way, as a smoothing factor. Consider the case where the long-term mean is defined as a constant, e.g. the average value of 1 year of interest rate, and that the value of mean reversion, $\alpha$, is proportional to the distance of the current rate to the mean interest rate, $\theta$. Then the changes of $\alpha$ would be augmented, since the more the rate would move away from the reference point, $\theta$, the greater the discrepancies of alpha with respect to it would be. By that, it becomes more clear, that $\theta$ incorporates the sense of trend, since the real mean value of interest rate in the current period varies from the mean of the previous period.

Mean reversion of Ornstein–Uhlenbeck process

The Ornstein–Uhlenbeck process is a stochastic process that describes the velocity of a particle moving under the influence of some, negative to the movement, force, e.g. friction, while over time the process tends to move towards a long term mean. The Vasicek model is derived from this process, sharing the same SDE (3.9). The only change, essentially, is the way each parameter is interpreted.

Weather derivatives are the family of marketable instruments that are dependent on the change of weather conditions. They are generally used to hedge against severe or generally, unexpected weather in order to minimize losses. For example, high temperatures that destroy the crop yield. Their valuation is often done by a Black-Scholes type of model, however in [43], they are considered in terms of an Ornstein–Uhlenbeck process. Working with such products requires exhaustive investigation in order to recognize all the aspects that affect the underlying factors, in this case temperature, that will enable the practitioner to fit the data to such model. The
authors conduct an extensive analysis on daily temperature data, in their attempt to effectively de-seasonalize and de-trend them. This, to a great extent, is based on wavelet analysis, the outcome of which is later used to determine the number and the transformation type of the signal-input to an auto-regressive (AR) process. The ultimate goal is to determine a procedure that will be used as a reference point for the mean reversion estimation.

The de-trended and de-seasonalized data are input to a neural network that is trained to estimate the temperature value of the next day. Although not clearly defining the input variables or the neural network itself, they formulate a very interesting approach. The neural network, as the approximator of the next day temperature function $\gamma$, is incorporating the dynamics of the model without the explicit parameterization needed for the AR process. In this way, by calculating the derivative with respect to the input of the network, they yield the value of mean reversion. Formally, starting from a simplified discretized version of Ornstein–Uhlenbeck for $dt=1$, using the paper’s notation:

$$\hat{T}(t+1) = \alpha \hat{T}(t) + e(t)$$
$$\hat{T}(t+1) = \gamma(\hat{T}(t)) + e(t)$$

where $\hat{T}$ denotes the pre-processed temperature data at time $t$ and $e(t)$ the differential of the stochastic term. Then computing the derivative, we calculate the time-dependent mean reversion value:

$$\alpha(t) = \frac{d\hat{T}(t+1)}{d\hat{T}(t)} = d\gamma/d\hat{T}$$

(3.11)

Approximating the calibration procedure as a function

Neural networks have a set of properties that enable practitioners of various fields to adapt them, one of them is speed. The time-consuming training process can be done off-line, while the forward run, the actual use of the network, is done with significant speed even for complex data and deep architectures. In many cases, time consuming MC simulation are employed to determine the most suitable value of a model parameter, similar to mean reversion for Hull-White model. The speed advantage of neural networks over other computationally expensive methods, inspired the author of [26] and [27] to bring their learning capacity to the calibration process. Although in both papers the neural networks are used as function approximators, it is worth studying this work in order to get a clear understanding of how the calibration procedures could be approximated by neural networks. The author offers a clear explanation of his experiments and the network architectures used, while he describes how NNs can be exploited alongside ready-made quantitative analysis libraries.

Consider Levenberg-Marquardt and Jamshidian decomposition that were discussed previously. A typical Hull-White calibration in a quant library is done by a combination of the two; First, the initial values for the volatility and mean reversion parameters along with interest rate curve data and volatility calculated by Jamshidian decomposition are supplied to Levenberg-Marquardt algorithm. Then, the optimization procedure determines the values of $\alpha$ and $\sigma$ that yield the lowest error for the current instruments. This procedure is approximated for one-factor Hull-White in [26] with a neural network yielding results relatively close to that of the optimizer. In the second paper [27], the same technique is used to approximate two-factor
Hull-White model.

The approach, although simplistic, offers a solution for certain challenges; the existence of relatively few data is addressed with a statistical procedure that creates new data points from the existing dataset, based on the errors of the calibration. On this data, a nine-layer feed-forward network is trained to map the input, yield curve and market volatility, into mean reversion and volatility parameters. From the data science perspective, forging data from an existing dataset is generally not preferred or even acceptable, but the author claims that it can work. However, this method has the risk of creating data that eventually lead to overfitting, and consequently sacrificing the ability of the model to generalize. Apart from that, this algorithm is not capable of preserving the relation between data points, it practically treats time series as independent points data. This is not a problem for the approach followed in these papers, but it is not in line with our initial view on calibration, so we will not use such an algorithm. In the code base of the paper the use of QuantLib (QL) solves all the practical issues that arise when working in this field, it demonstrates the use of a ready-made tool for calculating forward rates, apply day-count conventions and most importantly use out of the box all the calibration techniques we have seen in the previous section.
4 Mean reversion in the Hull-White model

We have seen that mean reversion calculation can be approached in different ways depending on the underlying model. In the Vasicek model, mean reversion parameter is assumed to be constant for certain period of time. This assumption is lifted in more generic versions, accepting it as a function of time. Under Hull-White, when calibrating with linear regression, \( \alpha \) is re-calculated periodically on historical data as the day-to-day changes are not significant for sufficiently long historical data. This approximation results in values within the interval \((0.01 - 0.1)\) \[17\], but often this is violated extending the upper bound \[28\]. It is common among practitioners to set alpha by hand, based on their experience and current view of the market.

4.1 Hull-White solution

Consider the generic Hull-White formula with all \( \theta \), \( \alpha \) and \( \sigma \) being time-dependent:

\[
\frac{dr(t)}{dt} = (\theta(t) - \alpha(t)r(t))dt + \sigma(t)dW(t) \tag{4.1}
\]

Applying Ito’s lemma for some \( s < t \) yields:

\[
r(t) = E(t,s)r(s) + E(t,s) \int_s^t E(u,s)\theta(u)du + E(t,s) \int_s^t e^{\int_v^u \alpha(v)dv} \sigma(u)dW(u) \tag{4.2}
\]

\[
E(t,s) = e^{-\int_s^t \alpha(u)du} \tag{4.3}
\]

The discretized form of equation (4.1) for \( dt = \delta t \) is:

\[
r(t+\delta t) - r(t) \approx (\theta(t) - \alpha(t)r(t))\delta t + \sigma(t)\sqrt{\delta t}
\]

\[
r(t+\delta t) \approx \theta(t)\delta t + r(t)(1 - \alpha(t)\delta t) + \sigma(t)\epsilon(t) \tag{4.4}
\]

where \( \epsilon(t) \) denotes a value sampled from a Gaussian distribution with mean 0 and variance \( \delta t \).

To avoid notation abuse, this discretized approximation will be written as an equation in the following sections. Hull defined the expression for \( \theta \) with constant \( \alpha \) and \( \sigma \) as:

\[
\theta(t) = F_t(0,t) + \alpha f(0,t) + \frac{\sigma^2}{2}(1-e^{-\alpha t}) \tag{4.5}
\]

where \( F_t(0,t) \) is the derivative with respect to time \( t \) of \( f(0,t) \), which denotes the instantaneous forward rate at maturity \( t \) as seen at time zero. Similar to \[24\], in our dataset the last term of the expression is fairly small and can be ignored.

\[
\theta(t) = F_t(0,t) + \alpha f(0,t) \tag{4.6}
\]

By omitting this term, the calibration of mean reversion becomes simpler since the calculation of the volatility term is not needed.

The general formula for the forward rate using the notation of \[24\] p.99 (5.2) reads:

\[
R_F = \frac{R_2T_2 - R_1T_1}{T_2 - T_1} \tag{4.7}
\]

where \( R_1, R_2 \) are the rates for times \( T_1, T_2 \) respectively. The instantaneous forward rate in terms
of zero rate can be approximated:

\[ f(0, t) \approx -\frac{\ln Z(0, t + \Delta t) + \ln Z(0, t)}{\Delta t} \]  

(4.8)

where \( Z \) denotes the zero rate derived from \( Z(0, T) = e^{-\int_0^T Y(u) du} \), where \( Y \) is the yield curve.

\( F_t(0, t) \) can be approximated by finite differences as:

\[ F_t(0, t) = \frac{\partial f(0, t)}{\partial t} \approx \frac{f(0, t + \Delta t) - f(0, t)}{\Delta t} \]  

(4.9)

These values can be easily calculated so that we can compute \( \theta \) for given \( \alpha \) and \( \sigma \). Combining with (4.4) for \( \delta t = 1 \) we have:

\[ r(t + 1) = F_t(0, t) + \alpha f(0, t) + r(t)(1 - \alpha) + \epsilon(t) \]  

(4.10)

### 4.2 The cyclical theta problem

In the case that calibration does not precede the calculation of long term mean, we face a cyclical problem. To address it, mean reversion can be calibrated by disregarding the existence of \( \theta \), implying that \( \alpha \) is directly related only to the movement of interest rate, similar to linear regression approach. Alternatively, the model is calibrated based on the initial expression, preserving \( \theta \), similar to LM approach. This, however, can lead to abrupt day-to-day changes of mean reversion and overfit to the volatilities of the current instrument. This behaviour can be observed in figure 4.1 on our test dataset.

![Fig. 4.1: Calibrated parameters with Leveberg-Marquardt](image)

Equation (4.5) stems from the algebraic expression of Hull-White, defining the theoretical long term mean in terms of the model. Note the presence of the forward rate, based on Hull’s indication [44], which translates into that the expectations of the market should be used to define the mean of the current period. Essentially, the simplification of (4.5), equation (4.6), implies that the rate will generally follow the initial forward rate curve and in case it deviates it will return back to it with speed \( \alpha \).

In figure 4.2 we observe \( \theta \) curves in two instances of our test dataset. The mean reversion used for these calculations was calibrated with linear regression on 100 historical data points.
Our observations align with the experiments in [28], the calculated $\theta$ follows similar levels. Notice the small discrepancies in the first part of the curve created by the surplus of data points in short maturities. This is caused by the approximation of the forward prime; in an attempt to minimize this issue, a slightly greater interval between the two sampled points of the function was used in the finite differences formula. Preserving the theoretical soundness for a sufficiently small $\Delta t = 0.0001$ we can rewrite (4.9) as:

$$F_i(0, t) = \frac{\partial f(0, t)}{\partial t} \approx \frac{f(0, t + \Delta t) - f(0, t - \Delta t)}{2\Delta t}$$

(4.11)

In figure 4.3 the influence of mean reversion on the calculation of $\theta$ is confirmed, since its value can significantly affect the long-term mean level, especially for longer maturities. The effect grows proportionally to the level of the value, changes within the interval $[0.0001 - 0.01]$ are clearly increasing the level of $\theta$ much less than in interval $(0.01 - 0.2]$.

Fig. 4.2: Interest rate curves for GBP Libor

Fig. 4.3: The effect of mean reversion parameter on long-term mean
5 Calibrating with neural networks

Calibrating mean reversion based on the current asset volatility leads to the twofold overfit, as discussed in the context of consistency hints 3.2. For that, as previously mentioned, strategies that rely solely on the interest rate term structure have been developed. Such methodology, using neural networks, is formulated in this section. The structure of the interest rate data is discussed, and the pre-processing steps that enable our datasets to serve as input to the proposed models are explained.

5.1 Studying the data

In our analysis, data from three different markets is used, curves based on Eonia index (EUR swap rates), 6M GBP Libor index bootstrapped on top of overnight swap rate, and swap rates based on 3M USD Libor.

Our GBP data consists of 891 time-points from January 2, 2013 until June 1, 2016 with 44 maturities ranging from 0, 1, 7, 14 days, 1 to 24 months, 2 to 10 years and 12, 15, 20, 25, 30, 40, 50 years. The curve has been bootstrapped on top of OIS and only FRA and swap rates have been used. This dataset is used in [26] and in [27].

The first characteristic to notice in figure 5.1 is the almost identical movement of rates with different maturities. This phenomenon is created by the way the yield curve is constructed. For short maturities the underlying interest rate is used, i.e. Libor 1M 3M 6M and 12M which reflect the zero-coupon rates, the mid-range maturities are based on FRAs and the longer maturities by swap par rates derived from the market [45]. In order to put these rates in common form they are going through the bootstrapping procedure; starting with the shortest maturity deposits, the rate is converted into a discount factor, then move forward through the available rates using the previous rates to calculate discount factors through time.

However, we can see parts of the graph that the rates grow closer together or further apart, this translates into steep or more gentle yield curves. Apart from the dependence of the current/previous rates to future rates, the relation we attempt to learn is when and if these rates are going to move towards each other. Since the yield curve is seen as an indicator of the current market condition, rates of different maturities growing closer reveals an uncertain market, notice
this phenomenon in EUR (fig 5.2) and USD dataset (fig 5.3) in the period of the economic crisis of 2007-2008. Gradually the yield curve was flattened and partially inverted on a generally high level. The cost of short term lending was as high as the cost of long term, which is deviating from the “norm” that longer term repayment has higher risk, which is expected to be compensated by the corresponding profit.

![Fig. 5.2: European swap rate per maturity](image)

Notice in figure 5.1, the aforementioned surplus of data points in the short maturities that make $\theta$ calculation unstable; the slope in this part of the instantaneous forward rate curve is very densely sampled, that results to an almost flat curve of which the approximation of the derivative becomes volatile.

![Fig. 5.3: USD Libor per maturity](image)

The EUR dataset is comprised by 3223 data points from July 19, 2005 until November 22, 2017 with 22 maturities ranging from 1-12, 18 months, plus 2-11 years, while USD dataset starts from January 14 2002 until June 6 2011, 2446 data points with 16 maturities 1-10 and 12, 15, 20, 25, 30 and 40 years. Our datasets are fairly limited as the recorded values are daily, and in order to include all the existing rate regimes, we restricted the available maturities to 22 and 16. The selected maturities are available for the greatest part of historical data. The EUR dataset starts from 2000, however, longer maturities were recorded gradually through years. The starting date
in 2005 was chosen as a good trade-off between number of maturities and observed regimes. Note that for EUR currently 30 swap maturities are recorded, up to 50 years, starting from 2011.

We are evaluating our approach in comparison to certain calibration techniques currently used, one of which is implemented in Quantlib, and in the context of Hull-White model, it requires swaption volatilities. As the GBP-Libor dataset is our test dataset, we use log-normal swaption volatility quotes for the respective dates with option maturities ranging from 1 - 10, 15, 20 years and swap term from 1 - 10, 15, 20, 25 years.

For our models, sigmoid/tanh and relu activation functions were tested, all perform nominally when the data is standardized. Standardization is required, especially for the sigmoid family of activation functions, since it ensures that the input will not be trapped in the valleys of the function. However, we have to go a step further and underline that for architectures handling input series independently, at least in the first layers, overall standardization, i.e. standardize all terms (maturities) together, will cause some issues; it will minimize our ability to transfer knowledge from one dataset to another as it will create strong dependence to the upper and lower bound of the underlying training set. It will also blur the widening and narrowing movement of the rates, making the series again, highly dependent on the movement of the longest and shortest maturities. For these reasons, we have used per term standardization, ensuring the preservation of the vertical movement between maturities and decorrelate each maturity series in the dataset from the rest. For similar reasons, normalization has been avoided altogether to keep the upper/lower values of each term unbounded.

5.2 Neural network with respect to Hull-White’s mean reversion

Our main approach for mean reversion calculation, is based on the assumption that historical rates can explain the future movement incorporating the sense of long-term or period mean value. The evolution of the yield curve is exploited in order to learn latent temporal patterns. This is achieved by training a generic function approximator that learns to predict the next-day interest rate. Starting from Hull-White model (eq. (2.4)), by discretizing similar to (4.4) yields:

\[
\frac{dr}{dt}(t) = (\theta(t) - \alpha r(t))dt + \sigma(t)dW(t)
\]

\[
r(t + \delta t) = \theta(t)\delta t + r(t)(1 - \alpha\delta t) + \sigma(t)e(t)
\]

Let \(\delta t = 1\) and \(\kappa = (1 - \alpha)\)

\[
r(t + 1) = \theta(t) + \kappa r(t) + \sigma(t)e(t)
\]

We train a neural network with input \(r\) as the function approximator to learn the generalized version of (5.2) expressed as:

\[
r(t + 1) = \gamma(r(t)) + e(t)
\]

where \(e(t)\) denotes the measured error. Then by calculating the derivative with respect to the input of the neural network, we can compute the values of the time function \(\kappa(t)\) as:

\[
\kappa(t) = \frac{dr(t + 1)}{dr(t)} = \frac{d\gamma}{dr}
\]
Neural networks are constructed to be fully differentiable functions and their derivation is a well defined procedure, which can be found in [46].

5.2.1 Can theta be replaced?

Notice that in equation (5.3), \( \theta \) is not an input of \( \gamma \), under the assumption that since the long term mean should be based on the current and historical interest rates, only them are needed as input to the network, which will infer the implied relations. In terms of [43], by not including \( \theta \), the equivalent preprocessing of the data is omitted. However, under the Hull-White model long term mean relies on the forward rate which describes more complex relation between \( r(t + 1) \) and \( r(t) \) that is not modeled explicitly in the network. Removing this information from the neural model makes the approximation to have a looser connection to the assumptions of the Hull-White model; in principle, it breaks the mean reverting character. To address that, we will describe two different approaches along with equations (5.2) and (5.3). The first incorporates \( \theta \), calculated on top of linear regression-calibrated alpha, resulting to:

\[
\hat{r}(t + 1) = r(t + 1) - \theta^{LR}(t) \tag{5.5}
\]

This approach could be accepted on the grounds that the mean reversion value used, is a good approximation and only partially affects the value of long-term mean. However, relying on a different model, such linear regression, introduces limitations, forbidding the development of a standalone method that follows specific assumptions. Avoiding the use of existing solutions ensures that the proposed models do not inherit any negative or positive properties and are not bounded to their specifications. For these reasons, this approach is not explored in the scope of this work.

As previously mentioned, the prevailing feature in the calculation of a long term mean is the forward rate. By keeping this information, the theoretical dependence of the movement of interest rate to the market expectations is partially embodied in the training data. This is realized by subtracting the first term of \( \theta, F_t \) as expressed in equation (4.11), to offer an approximation to long term mean value:

\[
\hat{r}(t + 1) = r(t + 1) - F_t \tag{5.6}
\]

Using equations (5.5) and (5.6) the neural network is used to learn the evolution of \( \hat{r} \):

\[
\hat{r}(t + 1) = \gamma(r(t)) + e(t) \tag{5.7}
\]

5.3 Models

As a first exercise we develop a depth-wise convolution based architecture to approximate the default calibration process built in QuantLib. Several properties are expected to come up; first a smoother mean reversion slope with the absence of jumps, and second, an imperfect match of the underlying parameters as a result of the limitations of a simple/shallow network. Then, we build our neural networks based on recurrent and convolution layers for mean reversion calibration and conduct a series of experiments to study the effects of the length of historical input and determine the effectiveness of the approaches discussed in 5.2.1 to the Hull-White model. In these experiments, we are going to study the possibility of market-knowledge transfer, i.e. how
effective a model trained with data of one market (e.g. EUR) could be, in calibrating on a different dataset (GBP).

There are various parameters that can be tweaked to improve the training and the general outcome of a neural network, however, in our experiments we focus on the effects of the parameters described in the previous paragraph that are considered substantial. Conducting technical tests in order to study, for example, the effects of various regularization methods or the differences caused by various batch sizes, is not a part of this work. In general, the training parameters of the proposed networks are configured, based on empirical knowledge and results, in the following way:

- Optimizer: SGD
- Initial learning rate: 0.01-0.035
- Learning rate decay: exponential 0.9 - 0.98
- Weight regularizer: l2
- Regularizer strength: 0.001
- Weight initializer: Xavier
- Loss function: Mean squared error
- Evaluation set percentage (fixed split): 20%
- Framework: Tensorflow

5.3.1 Convolution architecture

![CNN architecture](image)

Fig. 5.4: CNN architecture

In principle, convolutional modules are able to learn patterns from time-series (temporal) or images (spatial) preserving translation invariance, i.e. a learned pattern could be recognized regardless the change in size or position. This property proves architectures that rely on convolution beneficial for prediction tasks. Studying the data, it was pointed out that a very strong correlation between different interest rate maturities exists. A strictly data-based approach would be to simplify the datasets to few time-series that would preserve the temporal patterns. However, this would prevent the neural models from learning spatial patterns, the parallel movement of rates, dispersing or gathering. In order to effectively extract both characteristics of our data, a depth-wise convolution module is employed, which applies convolution filters at each
channel separately and then concatenates the resulting features. The correlation is left to be learned by the following dense layers (Appendix Architectures).

In our tests, the convolution architectures are made up from a depth-wise module followed by three dense layers in the form of e.g. 50-20-44 nodes, depending on the number of input signals. The number of dense layers was specified by experiments, the results of which showed that deeper architectures achieve worse accuracy, given the limitations imposed by the size of the datasets.

5.3.2 Recurrent architecture

![Fig. 5.5: LSTM architecture](image)

The recurrent architecture is a four layer neural network, with two LSTM layers followed by two dense layers. The selection of LSTM modules is made for their ability to catch temporal patterns both of long and shorter periods. They are widely used to learn sequential data and preferred over simpler or other modules of the RNN family, because of their robustness. While CNNs are able to extract features from data, that enables the identification of temporal patterns, LSTMs identify such patterns by capturing sequential dependencies in historical evolution. Their popularity is obvious in the literature, counting a large number of publications and diverse applications [47][48][49][50][51].

5.4 Mapping

In equation (3.4) mappings from linear regression parameters to Hull-White were provided. Similarly, the calculated values from neural networks need to be transformed to be usable within the interest rate model. From equation (5.4) let \( \frac{dr(t+1)}{dt} = \frac{dr(t)}{dt} = Y(t) \). Beginning with the simple discrete form of the Hull-White model (eq. (4.4)) as expressed in equation (5.4) we get:

\[
Y(t) = \kappa(t) \\
Y(t) = 1 - \alpha(t) \\
\alpha(t) = 1 - Y(t)
\]

(5.8)
The mapping in equation (5.8), is appropriate for networks accepting input only the rate of the previous time-step $\delta t = 1$. To be consistent with the initial assumption of multiple historical input, the mapping used should be suitable for functions regardless the distance of the sampled points. We move back to constant $\alpha$ and $Y$ for simplicity, and starting with the general continuous expression of Hull-White model, we insert equation (3.2) in our derivation:

$$Y = d \left( r(t) e^{-\alpha \delta t} + \frac{\theta}{\alpha} (1 - e^{-\alpha \delta t}) + \sigma e^{-\alpha (t-s)} \int_s^t e^{\alpha (t-u)} dW(u) \right) / \delta t \tag{5.9}$$

$$Y = e^{-\alpha \delta t}$$

$$\alpha = -\ln(Y)$$

The expression for $\alpha$ is the same as the one used for linear regression. The output of the networks define a partial solution for each historical time-step that is provided, i.e. the partial derivative with respect to each input, that is mapped to the time-dependent alpha function. In that sense, the output of the neural network is treated as the result of the procedure $Y(t)$:

$$\alpha(t) = -\ln(Y(t))/\delta t \tag{5.10}$$

### 5.5 Default calibration with QuantLib and linear regression

Neural network models will be evaluated in two ways, in comparison to linear regression and based on the error produced by LM optimizer, given the network’s calibrated mean reversion value. Traders are commonly using linear regression trained on past data to define mean reversion with the following procedure; having an extensive dataset of past interest rates, it is split in non-overlapping periods, e.g. 500 days, on which linear regression models are trained and then the average of the output of each one is used as the mean reversion. The consequence is a de facto constant mean reversion value, mainly serving the practitioner’s intuition for steady alpha rate, while it ensures the absence of negative values. However, this method lacks theoretical support, it is a strategy constructed empirically. For the purposes of this work, this approach is inadequate since we aim to model a time-dependent mean reversion which could not be compared with a constant value. In order to balance that and offer a more suitable, and, in a way, a more reasonable strategy, we apply the same calibration approach using linear regression with data used in a rolling window manner. Being limited by the extent of the datasets, the window size varies from 300, 400 to 500 time points. This approach is prone to negative alpha values, which cannot be used in Hull-White model, but offers a clear, unbiased view over the market condition. For this reason, instead of training a linear model, the Pearson correlation between previous-next data points is computed, so that the results of linear regression are approximated without yielding negative alpha values.

The second evaluation of neural networks’ results is conducted with QuantLib’s Hull-White process using LM optimizer. Since the networks are calibrating the mean reversion parameter, while the Hull-White process requires the volatility parameter as well, the calculated $\alpha$ from

---

\(^1\) $\delta t = 1$ stands for one time-step, in practice $\delta t$ varies depending on the day count convention used, for */365 conventions one day equals 0.0027
the neural model is inputted to the optimizer which calibrates $\sigma$. The error of the model is then comparable to the default calibration. Although this may be a suitable reference point for our purposes, the optimizer approach is not preferred for real world use, since the results can vary depending on the starting point, a notable example can be found in [26]. Additionally, as explained earlier, it relies on the volatilities of the underlying assets, which leads to overfitting to their values.
6 Results

The main assumption of the approach described in the previous sections, is that a neural network based on a series of past values should be able to predict future interest rates with acceptable accuracy, similar to existing linear models. It is also expected, not to suffer from extreme day-to-day variance as experienced in [43].

We expect that the use of the whole yield curve will maximize the informational gain from the market data. The speed of mean-reversion calibrated based on the movement of the whole yield curve, will follow the market changes more closely without being affected by high volatility in a parts of the yield curve.

6.1 Approximating the default calibration

Starting with a simple experiment, we attempt to approximate QL’s calibration with a neural network accepting the same input as the Hull-White process developed in QL, similar to [26]. Due to the limited data, a convolution neural network (CNN) of 3 layers (a convolution layer followed by two dense layers) is used, while its results are compared with the nine-layer dense network trained in [27], by feeding the network’s output to QL. This simple network trained on the real data, although does not achieve the best fit, is able to sufficiently learn the underlying function and produce comparable results.

As figure 6.1 indicates, the two networks can sufficiently approximate QL’s calibration yielding good results. Both networks have been trained from the beginning of the dataset, 01.01.2013, until 09.10.2015, and the rest is left as evaluation period. For training the feed-forward network (FNN), 200000 samples were used, created by the algorithm described in [26]. This leads to overfit, that is obvious after a very short period from the end point of the training set. On the other hand, the CNN yields close but worse results along the training set, with the exception of the first part, but relatively better error score on the evaluation part. Overfitting is avoided with greater success. We have to point out the fact that both networks in certain points yield better results than the Levenberg-Marquardt (LM) calibration, an explanation to that is given

![Fig. 6.1: Average volatility error](image-url)
considering that the objectives of LM optimization procedure is to achieve the best net present value (NPV). This, for certain instruments, is calculated based on the market volatility. As previously mentioned, correctly modelling the volatility surface (implied volatility) is the main interest for swaptions as it consist the main factor for its valuation.

6.2 Linear regression

As mentioned in 5.5, the window size of the linear regression approach affects the overall outcome. Models trained with shorter time windows show high sensitivity to small changes of the curve. In this section we provide the empirical evidence that led us to use 300, 400 and 500 points window sizes to serve as a reference to the comparisons with the neural models. The reader can find in the appendix section the average deviation results of the neural networks with respect to the results of the linear regression runs. This relative accuracy measure is the average of mean squared error along the calibrated speed of mean-reversion time series (Appendix Relative accuracy).

![Fig. 6.2: Linear models window size comparison EUR](image)

We observe the significant differences between the linear models. The four LR series seem to follow the same trends in certain periods. We can see that for less training points the outcome tends to be more volatile, leading to significant deviations. However, as we have previously discussed, there are not ground truth values for alpha that we could use as sole reference point, instead, we calculate approximations.
In practice, though, as pointed out in [17], a highly volatile speed of mean reversion is not usable for pricing. In these particular examples in USD, GBP and EUR datasets, the evolution of the calculated mean reversion peaks well over previously reported levels [28]. These results indicate that the history length tested in the proposed neural models (5, 15, 30) cannot be trustworthy when used in linear regressions.

Moving to speed of mean reversion calibration with neural networks, the models are put into test to calibrate alpha based on the current term structure. The procedure, as it is outlined in section 5.2, begins by training the neural networks to predict the next time step. Then, after we set up QL calibration, we apply the respective pre-processing pipeline and feed the interest rate data to the neural network. Then we calculate the gradient with respect to this data and the result is mapped to mean reversion value in terms of Hull-White model. Finally, we input the calculated alpha in the LM optimizer for the Hull-White process to calibrate sigma. This step is only done for our test dataset (GBP). For every dataset, we have calculated mean reversion using linear regression, as described in section 5.5, and compare it alongside the output of the
neural models. Regarding the predictive accuracy, neural networks’ results per term are very close to linear regression’s accuracy, reporting mean-squared error of $5.0 \times 10^{-6}$ levels.

Fig. 6.5: EUR Mean reversion calibrated by CNN trained with Euro data

Beginning with the EUR dataset, in figure 6.6 the evolution of mean reversion calibrated by the networks is plotted together with the one calibrated by selected linear models.

Fig. 6.6: USD Mean reversion calibrated by CNN trained with USD data

The CNN approach seem to be in relative accordance with LR trends, following some of the patterns but not always agree on the levels of mean reversion. While there are mismatches and lags, we can spot several similarities in the movement. The evolution of alpha from CNN-5 suggests that a shorter history length results to less flexible model, at least for CNNs. Here we have three cases, with 5, 15 and 30 input length. On the lower end this behaviour is quite obvious but moving to more historical data points, the changes become stronger, resulting to higher and lower levels.
In our tests, we have observed significantly different behaviours of CNNs based on the kernel_size/history_size ratio affecting the smoothness and the overall performance. The networks used here, have ratios of 2/5, 1/3 and 1/3 for CNN 5, 15 and 30 respectively.

On the LSTM side, there is not a similar smoothness adjustment tool, which enables us to study the effect of history length easier. In figure 6.8 we see that shorter and longer history affect the outcome, but the relative movement is in close sync. In comparison to the CNN networks the levels are generally the same. We observe parallel evolution that is consistent in major changes, the two significant jumps are followed by all three networks. The first major difference from the evolution of LR (2012), seems as if the movement of LSTM is an exaggerated jump of the respective LR and CNN slopes parts. In the second (2014), we can see that the peak of the NNs coincide with the peak observed in LR-300 only, similar to CNN-30 but does not follow its level.
Fig. 6.9: USD Mean reversion calibrated by LSTM trained with USD data

For the USD dataset both CNN and LSTM preserve the same characteristics, while CNN-5 yields results closer to LSTMs, mostly because of the first part of the dataset (2004-2006), where lower alpha than LR and CNN (15 & 30) is reported. For the rest, LSTMs are closer to LR-500 but follow the same trends with CNNs.

Fig. 6.10: GBP Mean reversion calibrated by LSTM trained with GBP data

Training on the test dataset (GBP) is a greater challenge since the training part is significantly smaller than EUR or USD. This is reflected by the LSTM networks that are producing very unstable alpha for longer input and achieve worse accuracy (Appendix Accuracy). On the contrary, CNNs preserve small frequent day-to-day jumps but the overall level agrees with the LR counterpart. Towards the end of the dataset, they report approximately the same level of mean reversion while shorter LR values are changing rapidly.

6.4 Using forward rate prime

In sections 4.2 and 5.2.1 we have discussed the role of $\theta$ in Hull-White and the significance of forward rate for its calculation. In the previous tests, the forward rate was not present in any form in the data. Following equation (5.6), we encode forward rate information by introducing
the prime term of θ. The CNN-30 network is trained on this data, its results are in the following graph 6.11.

![Graph showing mean reversion comparison](image)

**Fig. 6.11:** Mean reversion comparison of prime subtracted and simple trained CNN on GBP

The levels of alpha are closer to LR-500 and generally undergo smoother transitions than the simply trained CNN, but the movement is mostly parallel, seen more clearly in the full history figure 7.1. In section 7, we will revisit this relation based on results of section 6.6, to discuss this phenomenon from a different angle. The reason this method is not applied to EUR or USD datasets, is that the data consists of 22 and 16 points respectively, half the GBP-Libor data, forbidding the calculation of a good approximation of the forward term.

### 6.5 Market knowledge transfer

Our next challenge is to transfer the knowledge acquired from data of one market to the other. Since EUR and USD data is richer both in history and regime variety, we are using the respective maturities of GBP data as an input to EUR and USD trained networks (E & U-NN). There are a few difficulties to consider; networks trained on EUR or USD can only accept a subset of maturities, which cannot be scaled with the scalers that have been trained on the respective datasets. We also have to underline, that using maturities other than the ones used for training can lead to certain instabilities that will be discussed in section 7. The differences we observe in the plots of this section should not be interpreted as direct representations of the effects of each network to alpha calibration, but rather as the effects of using different subsets of maturities in combination with history input length.

This was confirmed by isolating the maturities that are common in EUR and USD datasets i.e. 1-10 years. The CNN-30 network was trained on these limited datasets and then GBP mean reversion was calibrated, resulting both nets to yield identical alpha values.
In figure 6.12, CNNs roughly follow the same slope and all experiencing opposite movement than LR from 09-2015. Here, the movement between the networks is parallel, and the inflexibility of CNN-5 that was previously noticed is not present on this part of the dataset. In reality, this behaviour is affected when using a subset of maturities since the input and the results are scaled differently (Appendix 8.1).

The equivalent experiment with U-NNs do not yield very different results, here it is more clear that CNN-30’s evolution is trending similarly to LR before 09-2015. In contrast to our expectations the combination of 30 historical data points with USD’s maturities result in the preservation of the downward movement of the last part of the dataset, while networks with shorter input follow parallel upward movement.

The LSTM networks trained on EUR and USD rates produce much smoother results than the GBP equivalent, because of their two dense layers. Their movement is consistent and relatively parallel to LR. Similar to CNNs, all three networks undergo rapid changes in the last part of the dataset. These instabilities are observed again earlier in the series (Appendix 8.2), the levels are generally lower than LR, but slope in parallel.
Fig. 6.14: GBP Mean reversion calibrated by LSTM trained on EUR dataset

We report that the translation of USD LSTM results required special treatment to match GBP levels. The reported levels of alpha were very low and needed re-scaling. Nevertheless, all networks are mutually consistent. We observe that levels of LSTM-5 are higher in these parts of the dataset which is not the case for the whole calibration history (Appendix 8.3).

Fig. 6.15: GBP Mean reversion calibrated by LSTM trained on USD dataset

6.6 Isolating initial rate

While we can extract information from the whole yield curve, the initial short rate (first yield maturity) is the most immediate metric and the theoretical target of Hull-White. Regardless the input size 22, 30 or 44 maturities the short rate could produce a consistent approximation of mean reversion with the current set-up.
Mapping solely the first maturity yields more stable results across both CNN networks with smaller deviations. The behaviour illustrated by LSTMs here, is in line with our expectations i.e. networks with less input would result in smaller changes, while networks with more inputs tend to follow a slightly more volatile movement. Both LSTM and the CNN networks in figure 6.16 are trained on GBP.

### 6.7 LM results

Calibrating the alpha value from the term structure with the proposed methodology lacks the ability to approximate volatility values that are necessary for interest rate models to value swaptions.
the consensus among traders is that mean reversion speed should not change experience big abrupt changes, which is one of the reasons to question the optimizer approach. Nevertheless, for the purposes of this work it is a good alternative tool for testing. Here, we present the results of CNN-30 network using the full term-structure and only the short rate. The complete list of implied volatility error graphs can be found in the Appendix LM-calibration.

Fig. 6.18: Implied volatility error of plain CNN-30 isolating short rate
7 Discussion

One of the greatest challenges of working with mean reverting interest rate models is to
determine a theoretically sound long term mean value. Its interpretation varies and is often
formulated in the mathematical terms of the corresponding model. In that sense, there is no
universal definition of the long term mean that can be used in all the models describing a
movement with mean reversion. In the Hull-White model, the expression for the long term
mean is based on the instantaneous forward rate, volatility and mean reversion, which have
model-specific denotations. The existence of mean reversion, whilst generally accepted, can be
weak or not present [52]. Nevertheless, the model is used with positive mean reversion, even if
partly trained linear regression suggests otherwise. That may be one of the driving factors for
the development of the procedure described in section 5.5. In the EUR dataset, a -500 history
points rolling window linear regression- would report negative alpha values for a brief period in
the beginning of 2009, the same time-point that CNN-30, LSTM-5 and LSTM-15 output almost
zero. Bearing this in mind, we will discuss neural networks modelling mean reversion and their
results, accepting that the procedures used in practice do deviate from theory or are constructed
to preserve theoretical eligibility but also reflect the empirical wisdom. The proposed approach
is based on theoretical terms and is not meant to encode empirical knowledge in that form, but
is evaluated in this scope as well.

Throughout our tests, CNNs achieved better test-set accuracy than LSTMs in all datasets,
which of course is not the main interest of this work, but affects the calibration procedure
overall and play a role in deciding which is the most efficient network. Depth-wise convolution
seems indeed to be suitable for parallel time-series. We observed a steady change in predictive
accuracy from shorter history CNNs to longer, among the three, the network with 30-step history
depth achieved the best results. However, this comparison is not simple since CNNs require
certain hyper-parameter tuning, which exceeds layer count or history depth, but concerns the
characteristics of the convolution layer itself; separable module, kernel size and pooling. While
we focused on kernel size effects, our early tests did not favour max-pooling nor separable
convolution, which led us to not further explore these options. CNNs seem to be affected
by input length and kernel size with regard to alpha calculation. 5-step networks with 2/5
kernel size ratio yield steadier results with small day-to-day changes, whereas 15 and 30 step
networks generally seem to be more flexible. Particularly, CNN-15 evolution is closer to CNN-30
movement, undergoing slightly less abrupt changes. CNN-30 is the best performing network, in
terms of accuracy, but also yields alpha values comparable to linear regression’s more consistently
among all datasets.

LSTMs did not follow exactly the same pattern in terms of predictive accuracy, all three
networks, regardless the history depth, resulted in approximately the same levels. This can be
observed in alpha calculations as well, where all three networks are mutually consistent and
tend to follow the same slope more closely. However, the outcome in the GBP trained networks
suggests that this is the case when they are sufficiently trained. Figure 6.10 shows that the
mismatches of the output of the networks may increase with more input points.

Using the forward prime in the training phase yields smoother results, but generally on
the same levels as its CNN-30 counterpart. Observe that, overall, both networks follow the
trends of linear regressions, but CNN-Prime is closer and undergoes smoother changes, while
big jumps are absent (fig 7.1). This is an indication that forward rate, and market expectations
as an extension, can be indeed used to extract knowledge from the market and confirms Hull’s indication to use the forward rate.

Fig. 7.1: CNN calibrating GBP-Libor

When the initial (short) rate is isolated to calculate alpha, small level difference appears between simple and prime networks. This difference is consistent all along GBP data and the relative movement is parallel. In section 4.2, we studied the movement of $\theta$ based on forward rate and pointed out the abnormalities in the beginning of the slope caused by dense sampling that makes forward rate resemble flat line similar to [28]. The first point of the slope, the short rate, undergoes such small changes, resulting to small values when approximating the prime, at the level of 1-e5, consistently all along the dataset. This creates a limited but steady shift of the slope. Likewise, the inclusion of forward prime prevents CNN from a steep climb in the period 04-2013 to 08-2013, in contrast to the plain CNN in figure 7.1; the steep upward slope of the term-structure results to greater prime values that act as a normalization factor. This effect is proportional to the steepness of the curve.

The most interesting experiment was to test networks trained on EUR and USD data, on GBP. The relative movement of the calibrated value was following the expected patterns, equivalent to the convolutional network trained on plain and prime-subtracted GBP data. Notice that the jumps of CNN-30 and LSTM networks in the last part of the dataset (fig 6.14), are also observed in the LM calibration, but are not seen in networks trained on GBP. Generally though, CNN-30 trained on EUR can be compared to the CNN-Prime. The equivalent experiments with LSTM networks yield generally smooth results, but in an altogether lower level, which indicates that LSTM should to be treated differently. The outcome of U-CNN indicates, again, that the relative movement of maturities does affect calibration, but more importantly, confirms the need to use the whole term structure to yield better informed mean reversion values.

Starting with the GBP dataset (fig 5.1), we will attempt to explain the movement of alpha with regard to the actual movement of interest rate, using as reference points CNN-30 GBP, CNN-Prime GBP and CNN-30 EUR. The CNN-30 USD is omitted, since the limited extent of USD maturities, together with the absence of less than 1 year maturity, influence the results greatly.

In the beginning, we observe an upward movement, shaping a hump, in both GBP and EUR CNN networks trained with plain interest rate data, in the region where mid and higher
maturities are moving upward, while shorter maturities remain compressed. This hump is observed only in plain CNNs (EUR and GBP), since they are affected by the movement of shorter maturities, that in this period undergo an inversion. This observation confirms the intuition that the mismatch between prime and plain networks is caused by the absence of weighing the effects of certain parts of the curve. Indicatively, in this period (04 to 08 of 2013), the hump is caused by the (inverted) movement of very short maturities. E-NNs use data up to 10 years, so the effect of this movement is augmented, while the absence of higher maturities from the input of the network prevents their normal movement to influence the outcome. The GBP-trained CNN-30, though, uses up to 50 years, so this movement lessens the effects of the inversion in the lower part of the curve. Similarly, U-CNN does not create a hump because this inversion is only partially seen by the network, due to the limited input maturities on the shorter end. Notice that as soon as this inversion has faded out and the curve returned to be normal, both networks return to sloping similar to prime.

After 07-2013, we see a gradual expansion of short maturities followed by a relatively steady parallel movement of mid and higher ones. During this period all three networks (fig 7.1) report consistently steady downward slope of mean reversion. This is explained if one considers the mean value of interest rates for this period, which is influenced by the continuous steady levels of the rate, causing the mean reversion to slow down.

This situation changes from 07-2014, where a second hump is created in the movement of alpha, which corresponds to a change in IR that begins a downward sloping. Higher maturities are dropping analogously to shorter maturities, causing a uniform compression of the curve, unlike the first period. That is the reason, CNN-Prime is following the same trend as well, it is affected by the change, as all maturities undergo suppression, which means that the whole forward curve changes as well. However, in 08-2014 networks accepting higher maturities form a peak, whereas E-CNN stabilizes. Going back to the IR curve, we observe that higher maturity signals, exactly at the beginning of 08-2014, start their downward movement faster than the lower ones. This change is reflected to slightly higher mean reversion.

The following period is characterized by only limited instability, which is reflected by the dropping alpha. In 04-2015, we observe CNN-30 Prime and CNN-30 EUR to agree on a lower level, but plain CNN to be more sensitive to the small hump of the IR evolution. Then, all three series experience a parallel upward trend, that is stabilized between a relatively balanced phase from 08-2015 to 01-2016, which then leads to the volatile part of the series around 04-2016. The movement, here, can be explained by the previous behaviours. We expect the prime network to follow steady high levels of alpha since the whole curve is getting compressed, while there are some small inversions. Plain CNN is expected to move upwards similar to the initial hump, reflecting the effects of the inversions of higher maturities and exaggerate the compression of the curve similar to period (12-2014 to 05-2015). Mean reversion calculated by E-CNN shows a big drop around 02-03 of 2016, which coincides with a parallel small drop of IR. This is seen in CNN-Prime as well, but in a smaller scale. Notice that the lower maturities in this period are not inverted but compressed, and later they undergo compressing and widening movements in contrast to the steadier higher maturities. This explains why E-CNN produces the most volatile signal in this part of the dataset.

Moving to the EUR dataset, first we should point out the similarities between the results of the two networks, LSTM and CNN in figure 7.2. There is relative agreement on certain parts of the slope. Both networks have floors and peaks that coincide on many points throughout

44
the time series. CNN-30 is more volatile and more sensitive to IR changes, while both report alpha values close to 1% in the initial period that is characterized by high and almost flat IR curve. This period is followed by a fast normalization in 2009, where networks report an upward movement similar to GBP dataset where big IR changes have the same effect on mean reversion. The same pattern is repeated in mid 2011, where IR increases uniformly. During 2012-2013 both slopes slowly drop, but preserve higher levels than LR. In the last parts of the time-line, the rates are stabilized but undergo small jumps, a situation that is reflected by the downward movement of alpha expressed by CNN and more mildly by LSTM. Particularly from mid-2015 to mid-2016, all LR alphas report a small jump which is caught only by CNN.

![Graph](image1.png)

**Fig. 7.2:** CNN and LSTM calibrating Eonia

The first period of USD, 2004-2006, is characterized by jumps in LR calibration in a relatively high level around 2%. While CNN is close to the behaviour observed in the other datasets, i.e. moving downward when IR tend to stabilize, LSTM reports significantly lower mean reversion, starting to closely approximate LR-400 from mid-2005.

![Graph](image2.png)

**Fig. 7.3:** CNN and LSTM calibrating on USD swap rate

Between the beginning of 2006 and mid-2006, we see all series moving downward followed by a fast climb by neural networks, which is explained by the change of the IR movement. For the period 2008-2009, CNN reports significantly different alpha than the rest, but behaves similarly.
to the same period in EUR dataset, regarding the slope. In 2009, both networks drop to almost zero followed by a steady low mean reversion as IR is stabilizing and moves upward when the signals of maturities spread.

Generally, the behaviour of the networks is consistent in all datasets. We see that the reported mean reversion moves upward when IR experiences fast changes. The sensitivity to these changes varies depending on the network and especially the number of maturities observed. Studying these results, we recognize the main advantage that the inclusion of forward prime offers; it lessens the sensitivity of the network to partial curve changes, augmenting the importance of the curve as a whole. Even if the evaluation of these results cannot be exact, the CNN networks seem to have greater potential and proved more suitable for prediction and mean reversion calculation. However, in real life conditions the use of a smoothing factor and the inclusion of prime are deemed necessary. LSTMs may require more data for training, but produce consistent results regardless the number of historical data points supplied. These findings suggest that history depth can play greater role in the performance of CNN networks than LSTM.
8 Conclusion and future work

Main conclusions

In our literature review we showed the most common ways to calibrate mean reversion speed for the Hull-White model, identifying certain shortcomings. We develop a technique to address these limitations by calculating the theoretical time-dependent mean reversion. This approach is based only on interest rate data, disregarding dependencies to the price of the assets. Discussing the findings of [43], we argue that a neural network that relies on more than one past values can approximate mean reversion while minimizing undesired fluctuations. This proposal is examined with a series of network architectures.

Addressing the first research question, the neural models were trained to predict the future value of interest rates. We underlined the importance of the full term-structure and crafted our networks to exploit the relative movement of different maturities. In our initial approach, the dynamics described by the Hull-White model were reproduced implicitly, by training the networks on plain interest rate data. By following the mathematical expression of Hull-White, that suggests the incorporation of forward rate into the long-term mean, we included the forward prime in the training dataset. This enforced the neural networks to be more consistent with the model’s assumptions. It contributed to a better fit to market changes, while mean reversion level and evolution were aligned more closely to the empirical expectations. This demonstrates how the findings and definitions of explicit computational models can be used effectively with neural networks.

The studied methodology, based on neural networks, outperformed algorithms based on linear regression with short-length input windows. More specifically, networks with one month of historical data input, yielded results significantly better than the equivalent linear models. In order to implement our approach, we employed recurrent and convolution based architectures. At the same time, we provided a suitable mapping from the derivative of the neural network to the Hull-White model parameter. Throughout our experiments, the characteristics of each network were explored, identifying distinct properties that follow each module.

In an attempt to address the shortage of data and enhance the versatility of the neural models, we created a procedure to map and pre-process interest rate series of multiple maturities. This procedure enabled networks trained on EUR (22 maturities) and USD (16 maturities) to be used for the calibration of the equivalent series of GBP data (44 maturities). This allowed the transfer of knowledge acquired from one market to another, without being limited by the availability of market-specific data. With the two network architectures, the second research question is addressed in two distinct ways. In the context of CNNs, the features learned from the high dimensional interest rate data were successfully deployed to calibrate the speed of mean reversion of a different dataset. In the same experiment, the sequential dependencies learned from the LSTM-based architecture resulted to smoother mean reversion evolution, generally similar to that of CNN.

A significant property of this approach is that the need to discard or treat historical data in parts is eliminated. All patterns are learned without significantly affecting the temporal relations that are already captured. In this regard, splitting data into time regions based on current market condition or regime changes do not benefit or protect the approximator from mistakes. Omitting specific sufficiently small regions of data, produces practically the same
results as when the full dataset is used for training.

The CNN architectures are favoured over LSTM ones for several reasons. Their predictive accuracy was greater in all datasets, they performed better in knowledge transfer and had no need for special treatment. The calibration results of CNNs were less smooth because of the existence of more dense layers. However, the observed behaviours were consistent with the movement of interest rates regardless of the market or the number of maturities.

Using neural networks to estimate measures introduced based on expert knowledge, as hidden variables in high dimensional data, constitutes an example of how strong estimators can improve explicit computational models. Calibrating the speed of mean-reversion in the context of the Hull-White model, also indicated that deploying features of elaborate processes can significantly enhance data-driven approaches.

Future work

While mean reversion is used in many interest rate models, there are differences in basic assumptions that explain its behaviour in the scope of each model and most importantly the way long term mean is defined. In this work the focus was Hull-White model but a step forward would be to extend this approach to other models describing mean reverting movement. This way the suitability of the mapping based on Hull-White could be confirmed or disproved. Such a study would also offer a stable ground for the way that time-dependent mean reversion should be treated and calculated from a single source (interest rate). It would provide more empirical evidence regarding the non-parallel calibration. Results similar to current work’s or [43] will enhance the use of time-dependent speed of mean reversion calibrated only on interest rate data. More importantly, strengthening this approach will offer a good alternative to strategies that allow both mean reversion and volatility to be calibrated simultaneously, based on the best fit to the current prices.

We discussed the difficulties to handle long term mean calculation in the way it was defined by Hull. The effects of incorporating the forward prime in interest rate data indicates that, explicitly including \( \theta \) in the training datasets may increase the accuracy of the calibration of mean reversion. The main factor that moved us away from this option is the need to rely on other techniques that will eventually affect the result of the procedure. We suggest to put together an algorithm that would continuously re-calculate \( \theta \) along with the network training process. Its value will be based on the speed of mean reversion calibrated by the network. This entails that \( \theta \) would be incorporated into interest rate data similarly to forward prime (section 4.2).

In the first training steps alpha will inevitably be calculated by an existing procedure, i.e. linear regression or the models trained in this work. Over time, while the network would be sufficiently trained, \( \theta \) would be re-calculated based on alpha values produced by the network itself. Essentially, we are describing a procedure that would try to approximate a changing function. This can lead to instabilities if small changes of alpha affected long term mean dramatically. Allowing only small changes can be enforced by a learning rate to limit alpha changes.

In a more technical note, an interesting direction would be to investigate approaches that neural networks will learn to predict more than one days in the future, even up to the life time of a financial contract. This will give practitioners a better view over the future, allowing the averaging that could result in less volatile alpha value. However, the most immediate solution to bridge the gap of theoretical result, to usable mean reversion for pricing and avoid averaging
approaches, is to apply wavelet transformation to smooth neural network's input, similar to [37]. That will produce steadier mean reversion values, since small discrepancies that are reflected in the output of the networks will be minimized. Wavelet transformation is often used with time series prediction [53][54][55] and is suitable for interest rate data that undergo often and small changes.
References


Appendix

Calibration

Fig. 8.1: Full history GBP- CNNs trained on EUR

Fig. 8.2: Full history GBP- LSTMs trained on Eonia

Fig. 8.3: Full history GBP- LSTMs trained on USD
Fig. 8.4: Full history - GBP Mean reversion calibrated by CNNs trained on GBP- Absolute gradient

**Prediction accuracy**

<table>
<thead>
<tr>
<th>History</th>
<th>5</th>
<th>15</th>
<th>30</th>
</tr>
</thead>
<tbody>
<tr>
<td>EUR</td>
<td>1.61E-04</td>
<td>1.08E-04</td>
<td>8.18E-05</td>
</tr>
<tr>
<td>USD</td>
<td>1.02E-04</td>
<td>5.07E-05</td>
<td>4.92E-05</td>
</tr>
<tr>
<td>GBP</td>
<td>2.10E-05</td>
<td>2.01E-05</td>
<td>2.01E-05</td>
</tr>
</tbody>
</table>

Table 1: Average mean squared error CNN

<table>
<thead>
<tr>
<th>History</th>
<th>5</th>
<th>15</th>
<th>30</th>
</tr>
</thead>
<tbody>
<tr>
<td>EUR</td>
<td>3.53E-04</td>
<td>3.06E-04</td>
<td>2.96E-04</td>
</tr>
<tr>
<td>USD</td>
<td>2.83E-04</td>
<td>2.40E-04</td>
<td>2.35E-04</td>
</tr>
<tr>
<td>GBP</td>
<td>2.17E-05</td>
<td>2.24E-05</td>
<td>2.20E-05</td>
</tr>
</tbody>
</table>

Table 2: Average mean squared error LSTM

**Relative average accuracy wrt linear regression**

<table>
<thead>
<tr>
<th></th>
<th>EUR</th>
<th>USD</th>
<th>Avg.</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNN</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>4.07E-04</td>
<td>8.06E-05</td>
<td>2.44E-04</td>
</tr>
<tr>
<td>15</td>
<td>3.65E-04</td>
<td>7.68E-05</td>
<td>2.21E-04</td>
</tr>
<tr>
<td>30</td>
<td>3.75E-04</td>
<td>8.64E-05</td>
<td>2.31E-04</td>
</tr>
<tr>
<td>LSTM</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>3.82E-04</td>
<td>1.36E-04</td>
<td>2.59E-04</td>
</tr>
<tr>
<td>15</td>
<td>3.71E-04</td>
<td>1.07E-04</td>
<td>2.39E-04</td>
</tr>
<tr>
<td>30</td>
<td>3.41E-04</td>
<td>1.18E-04</td>
<td>2.29E-04</td>
</tr>
</tbody>
</table>

Table 3: Relative accuracy to LR-300
**LR-400**

<table>
<thead>
<tr>
<th></th>
<th>CNN</th>
<th>EUR</th>
<th>USD</th>
<th>Avg.</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>1.13E-04</td>
<td>4.21E-05</td>
<td>7.77E-05</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>1.97E-04</td>
<td>4.40E-05</td>
<td>1.21E-04</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>1.69E-04</td>
<td>5.67E-05</td>
<td>1.13E-04</td>
<td></td>
</tr>
</tbody>
</table>

**LSTM**

<table>
<thead>
<tr>
<th></th>
<th>CNN</th>
<th>EUR</th>
<th>USD</th>
<th>Avg.</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>2.29E-04</td>
<td>6.65E-05</td>
<td>1.48E-04</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>1.68E-04</td>
<td>5.88E-05</td>
<td>1.14E-04</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>1.47E-04</td>
<td>5.78E-05</td>
<td>1.03E-04</td>
<td></td>
</tr>
</tbody>
</table>

Table 4: Relative accuracy to LR-400

**LR-500**

<table>
<thead>
<tr>
<th></th>
<th>CNN</th>
<th>EUR</th>
<th>USD</th>
<th>Avg.</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>2.42E-05</td>
<td>3.45E-05</td>
<td>2.94E-05</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>1.68E-04</td>
<td>4.23E-05</td>
<td>1.05E-04</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>1.18E-04</td>
<td>6.11E-05</td>
<td>8.97E-05</td>
<td></td>
</tr>
</tbody>
</table>

**LSTM**

<table>
<thead>
<tr>
<th></th>
<th>CNN</th>
<th>EUR</th>
<th>USD</th>
<th>Avg.</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>2.04E-04</td>
<td>3.56E-05</td>
<td>1.20E-04</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>1.24E-04</td>
<td>4.00E-05</td>
<td>8.20E-05</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>1.11E-04</td>
<td>3.40E-05</td>
<td>7.25E-05</td>
<td></td>
</tr>
</tbody>
</table>

Table 5: Relative accuracy to LR-500

**LM calibration**

![Graph](image)

Fig. 8.5: LM calibration with CNN-5
Fig. 8.6: LM calibration with CNN-Prime

Fig. 8.7: LM calibration with CNN-30

Fig. 8.8: LM calibration with EUR CNN-30
Fig. 8.9: LM calibration with EUR CNN-5

Fig. 8.10: LM calibration with EUR LSTM-15

Fig. 8.11: LM calibration with EUR LSTM-5
Fig. 8.12: LM calibration with USD CNN-5

Fig. 8.13: LM calibration with USD LSTM-30

Fig. 8.14: LM calibration with USD LSTM-15
Fig. 8.15: LM calibration with USD LSTM-5

Architectures

Fig. 8.16: Architecture Convnet
Fig. 8.17: Architecture Convnet Expanded

Fig. 8.18: Architecture Recurrent
Fig. 8.19: Architecture Recurrent Expanded