Learning Curve Dynamics with Artificial Neural Networks

Alexei Kondratyev

alexei.kondratyev@sc.com

Version: 11 April 2018

Abstract

Our objective is to learn the natural curve shapes with the help of Artificial Neural Networks (ANN). Our research aims to improve on curve dynamics predicted by PCA. By running the ANN on datasets of historically observed term structures of forward crude oil prices and interest rate swap rates we learn how the curves evolve over time. This allows us to determine stable natural shapes of the curves as well as to detect statistically significant deviations from the expected, predicted behaviour.

Keywords: Artificial Neural Networks, Autoencoders, Term Structure, Curve Dynamics

Key messages:

- Learning natural curve shapes and natural curve transformations with Artificial Neural Networks.
- ANN preserves more information extracted from the training dataset than PCA.
- PCA can be viewed as a limit case of the ANN with strong regularisation.
- ANN configured and trained as an autoencoder is able to detect atypical curve shapes.

JEL classification: C61, C63, G13, G15

1 Introduction

Risk management requires calculation of the probability distributions of market risk factors (in our case, curve shapes) over various time horizons. In the case of market risk management we are often dealing with short time horizons, for example, a 1-day or a 10-day VaR. In contrast, for counterparty risk it is necessary to know the probability distribution of the underlying risk factors over medium to long horizons, from 1-year for the internal model method for counterparty risk, all the way to portfolio maturity for limit management.

Modelling the probability distribution over a short interval from the historical time series data seems to be a simple enough task, for which there exists a large number of well established statistical models and techniques. Most of these techniques, however, are designed to model the probability distribution of returns, which we will generically define as a log change of a given market risk factor between the adjacent observation points. When modelling the returns, rather than the risk factor itself, the model would typically have no dependence, or a prescribed parametric dependence, on the initial level of the risk factor relative to which the return is calculated.

The assumption of homogeneity is practical and valid when there is an expectation of homogeneity with respect to the initial level of risk factor. For example, no stock price level is special (other than for the fleeting "support levels" we will not consider here), in which case the probability density of the returns can be modelled irrespective of the level of the risk factor itself.

The assumption of homogeneity which holds for most standalone asset-like factors, such as prices and FX rates, breaks down when risk factors are buckets of a continuous curve. The first way in which homogeneity assumption that holds for individual risk factor becomes invalid for curve buckets is that the probability distribution of returns for a given bucket depends on the surrounding buckets. This effect is typically modelled using the principal component analysis (PCA). The PCA technique identifies several leading harmonics in the shape of the curve which explain most of its variance. Then historical correlation and variance of these harmonics can be calculated.

The dependence of probability distribution of a curve bucket return on surrounding buckets is not the only way in which homogeneity breaks down. For most curves, including IR and commodity, the probability distribution also depends on the overall curve level. The dynamics of the IR curve is different at near zero rates compared to the dynamics at the rates level exceeding 10%. The dynamics of the oil futures curve is different when oil prices are around \$20/bbl compared to the dynamics when they are around \$80/bbl.

Traditional non-parametric statistical techniques for estimating the probability distribution of returns are ill equipped to deal with probabilities that nontrivially depend on the level of the underlying risk factor. This type of analysis is usually done with parametric models, which prescribe a stochastic equation driven by one or more stochastic drivers. The parameters of this equation are then calibrated to the historical data.

While the parametric model approach is more powerful in describing the differences in dynamics depending on curve level, such as the presence of mean reversion, its shortcoming is its reliance on a prescribed parametric model, as opposed to a model derived directly from historical data without any pre-defined notions of dynamics that the parametric model brings.

The problem of learning curve shape dynamics for medium to long time horizons gains another measure of complexity compared to the $t \to 0$ limit. As we go from short horizons to long horizons, the information about the initial state is progressively lost. In the limit of $t \to \infty$, the information about the initial state is lost completely, and the probability distribution is that the curve shapes measured over a long observation period

of a stationary process.

The limit when the information about the initial state is lost completely is rarely achieved in practice, as even at a 30-year horizon a considerable amount of information about the initial state remains. For the typical time horizons where PFE based limits are calculated, around 5 years to 10 years, the information about the initial state dominates, but the probability distribution can no longer be modelled in the $t \to 0$. To model PFE-based limits, we need to reformulate the problem such that we no longer deal with the returns, but instead deal with values (endpoints) of the stochastic process for the curve buckets. When expressed in terms of the risk factor values rather than their returns, it can be stated as finding the probability distribution of curve shapes at time $t = T$ given the initial shape at $t = 0$. Of course, we can only hope to measure such distribution accurately if the $t = 0$ shape relative to which we are calculating the probability distribution is well represented within the historical record we will use for calibration.

In this paper we will attempt to model curve dynamics using machine learning techniques, and specifically artificial neural networks (ANNs). We will show that an ANN consisting of just two hidden layers is able to calculate the probability distribution of curve shapes more accurately, and in a model free way, than other popular techniques.

2 Artificial Neural Networks

ANN is a network of interconnected activation units (neurons). Each activation unit performs three functions: summation of the input signals from all of the upstream units to which it is connected; linear or non-linear transformation of the aggregated input; sending result to the downstream units to which it is connected. In its simplest form the ANN is organised in layers of activation units: input layer, output layer and one or several hidden layers. Such ANNs are generally known as Multi Layer Perceptrons (MLP). The output layer often performs only aggregation (linear transformation). The hidden layers should perform non-linear transformations to give the ANN a chance to learn complex non-linear functions.

The practical approach to the ANN architecture is based on the fundamental result obtained by G. Cybenko in [1]. It states that arbitrary decision regions can be arbitrarily well approximated by continuous feedforward neural networks with only a single internal, hidden layer and any continuous sigmoidal nonlinearity. This result was further generalised to the wider range of activation functions by K. Hornik et al in [2]. They established that multilayer feedforward networks with only a single hidden layer and an appropriately smooth hidden layer activation function are capable of arbitrarily accurate approximation to an arbitrary function and its derivatives. In fact, these networks can approximate functions that are not differentiable in the classical sense, but possess only a generalised derivative.

Although an MLP with a single hidden layer is capable of learning complex functions, it is a general statement [3] that no more than two hidden layers are needed to approximate an arbitrary function with relaxed requirements for the type of activation function. In practice, there are several popular choices for the activation function that serve well in most cases: the logistic sigmoid function, the rectified linear unit function, and the hyperbolic tangent.

Results presented in this paper were obtained with the help of MLPRegressor, an MLP from the open source Scikit-Learn package of machine learning tools. An excellent overview of Scikit-Learn can be found in [4]. The following feedforward MLP network was constructed for the curve dynamics analysis:

MLP Parameter	Value
Number of units in the input layer	Number of curve tenors + 1
Number of units in the output layer	Number of curve tenors
Number of hidden layers	2
Number of units in each hidden layer	Number of curve tenors + 2
Type of activation function	Hyperbolic tangent

The motivation for this particular choice of parameters will become clear in the following sections.

It is a usual practice to split the full original dataset into a training and a validation datasets, typically in an 80:20 or a 90:10 ratio depending on the dataset size. The training is performed through the batch gradient descent algorithm (see e.g., [5]), which updates parameters using the gradient of the error function with respect to a parameter that needs adaption. The training process can be described by the following expressions:

$$\mathbf{w}^{(\tau+1)} = \mathbf{w}^{(\tau)} - \eta \nabla E(\mathbf{w}^{(\tau)}) \tag{1}$$

$$E(\mathbf{w}^{(\tau)}) = \alpha R(\mathbf{w}^{(\tau)}) + C(\mathbf{w}^{(\tau)}) \tag{2}$$

where $\mathbf{w}^{(\tau)}$ represents the array of network weights at the τth iteration of the gradient descent algorithm, η is the learning rate (speed with which weights are updated), $C(\mathbf{w}^{(\tau)})$ is the cost function (for example, sum of squared errors for the whole batch, i.e. for the whole training dataset), $R(\mathbf{w}^{(\tau)})$ is the regularisation penalty function, and α is the parameter through which we can control the degree of regularisation. The MLPRegressor was configured with the L2 regularisation, i.e. with the penalty function proportional to the squared norm $||\mathbf{w}||^2 = \sum_i w_i^2$.

3 Choosing the Datasets

We illustrate the application of ANN to the problem of understanding curve dynamics by using two curves, which are representative examples of their corresponding asset classes: brent crude oil forward price curve and USD interest rate swap curve. The choice is motivated partially by the central role these risk factors play in the financial markets and wider economy and partially by the availability of a long enough historical time series. The latter is particularly important as we work with the monthly absolute price (rates) returns.

The choice of the monthly frequency for crude oil forward prices is mainly driven by the following two factors. First, contracts with a longer time to maturity were traded less frequently in the first half of the period covered by the dataset. This means that we need to look at longer time intervals to determine meaningful curve changes. Second, the studies [6] show that monthly data has equal predictive ability to daily data. In the case of USD swap rates, monthly curve observation frequency allows us to eliminate significant amount of noise normally associated with more frequent observations and to make the trends more visible.

It is also important to note the well documented similarities between crude oil term structures and the shapes of the government bond yield curves (see e.g., [7] and references therein). In particular, the following stylised facts about the yield curves are also applicable to the crude oil term structures:

1) The average yield curve is increasing and concave.

2) The yield curve assumes a variety of shapes through time, including upward sloping, downward sloping, humped and inverted humped.

3) The short end of the yield curve is more volatile than the long end.

4) Long rates are more persistent than short rates.

Therefore, it is natural to run the ANN analysis on both crude oil and swap rates term structures. In order to use the same values of regularisation parameter for both datasets, i.e. to have both crude oil forward prices and swap rates at roughly the same level, all samples from the brent crude oil dataset were normalised by dividing all forward prices by 100. Once the network was trained and predictions were made, the results were multiplied by 100 to reinstate the normal scale.

4 Understanding Curve Moves with ANN and PCA

We start with the historical time series of a particular curve specified on K tenors $T_1, \ldots, T_K$. For example, we collect monthly snapshots of the forward oil prices specified on the fixed number of tenor points. From these observations we construct monthly curve moves. Our objective is to predict the most likely curve transformation given its observed shape at a particular moment in time. As was mentioned above, we expect that an ANN with two hidden layers will be sufficiently deep to forecast curve transformations. With respect to the rest of the ANN architecture parameters, they can be specified in the following way.

Let $P_T(t)$ be the forward price observed at time t for delivery in T months. Since we are interested in the non-overlapping monthly curve changes we set observation points t at calendar month ends. Let us use the following notation: $\Delta P_T(t+\Delta t) = P_T(t+\Delta t) – P_T(t)$, where $t+\Delta t$ is the next observation point of $P_T(t)$. Then the samples on which our ANN shoud be trained can be constructed as:

$$Sample(t) = [InputVector(t), OutputVector(t)] \tag{3}$$

$$InputVector(t) = [P_{T_1}(t), \ldots, P_{T_K}(t), \Delta P_{T_j}(t+\Delta t)] \tag{4}$$

$$OutputVector(t) = [\Delta P_{T_1}(t+\Delta t), \ldots, \Delta P_{T_K}(t+\Delta t)] \tag{5}$$

In other words, we want to train ANN to predict the one-period curve moves (from t to $t+\Delta t$) given curve shape at t and the magnitude and direction of the curve move at T_j tenor point. The T_j tenor move is our control parameter: we are interested in the overall curve shape conditional on the curve moving up/down at T_j tenor by x bps. This tenor can be chosen as the point of maximum variance. For example, it can be the tenor at which the first principal component reaches its maximum absolute value.

The ANN approach has an alternative in the form of PCA – another non-parametric model widely used for the analysis of curve dynamics. However, unlike ANN, which is a non-linear factor model, PCA is a linear combination of the principal components. These principal components are the eigenvectors of the covariance matrix that holds information about the curve moves at different tenors (the correlation structure and the magnitude of the moves). The eigenvectors are ordered according to the values of the corresponding eigenvalues: the first principal component is the eigenvector with the largest eigenvalue, and so on. The eigenvectors specify the types of possible curve transformations ("parallel

shift", "twist", "bend", etc.) and the PCA-predicted curve move is a linear combination of these independent transformations (with their corresponding weights).

PCA is a popular technique for explaining curve dynamics. But it is necessary to keep in mind that PCA relies on several strong assumptions, which are not always valid for an arbitrary dataset [8]:

1) Linearity. The PCA is, effectively, a change of basis and the key question it asks can be formulated as: is there another basis, which is a linear combination of the original basis, that best reexpresses our dataset? However, real complex dynamical systems are very often non-linear and the non-linearity may be responsible for the most interesting part of their behaviour.

2) Mean and variance are sufficient statistics. We assume that the mean and the variance entirely describe a probability distribution. The only zero-mean probability distribution that is fully described by the variance is the Gaussian distribution. Therefore, for this assumption to hold, the probability distribution must be Gaussian. Obviously, this is not always the case.

5 How the Curves Move

There are four main results which we illustrate below by looking at the two specific curves: brent crude oil forward price curve and USD interest rate swap curve. For both curves results are structurally similar, though they are more pronounced for the brent curve due to the fact that this curve has undergone massive transformations over the last 25 years.

(i) The properly trained ANN preserves more information about natural curve shapes and natural curve transformations than PCA. All of the information available to PCA is encoded in the covariance matrix. In contrast, ANN has the whole training dataset to analyse curve transformations conditional on the initial term structure.

(ii) In the limit of strong regularisation ANN converges to PCA.

(iii) Dissipation of information about the initial term structure after the curve is transformed multiple times (the long term horizon). Over a short time horizon ANN would predict curve transformations with a strong dependency on initial curve shape. But over a long period of time, ANN predicts curve moves between the natural curve shapes with almost no trace of the initial term structure.

(iv) ANN configured and trained as an autoencoder is able to detect atypical curve shapes.

5.1 Brent Forward Curve

The dataset is 25 years of monthly forward brent oil price observations (from March 1992 to March 2017) at the following tenors: 1M, 3M, 6M, 12M, 18M, 24M, 30M and 36M. From these observations we construct absolute monthly returns (a total of 300 curve observations). These returns are then used to train the ANN and perform PCA.

The PCA conducted on the absolute monthly forward price returns suggests that the 1st principal component ("parallel shift") can explain 94% of the variance and the 2nd principal component ("twist") can explain further 5% of the variance. The shapes of the first two principal eigenvectors show that the largest variance is observed at the 1M tenor (as shown in Figure 1. This and all subsequent figures are based on own calculations). Therefore, the 1M forward price is a natural choice for the curve driver, i.e. it is a natural

choice for the feature that indicates the magnitude and direction of the forward curve moves.

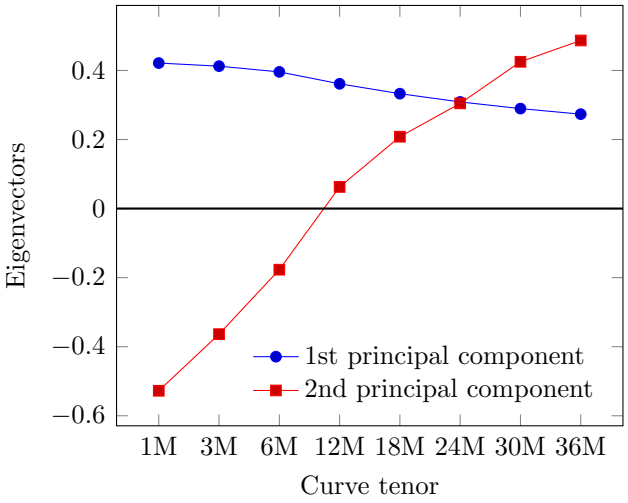

Figure 1: Principal Component Analysis

Once the ANN is trained on the dataset, we generate forward curve transformations driven by the controlled 1M forward price move conditional on the initial shape of the curve. The charts in Figure 2 show very different curve dynamics and strong dependence on initial curve levels for ANN trained with small value of the regularisation parameter.

Setting $\alpha = 0.00005$ (slightly below the MLPRegressor default value of 0.0001) ensures that we extract maximum information from the dataset by being less restrictive with possible values of the network weights. The overfitting in this situation is also avoided by having a sufficiently large number of samples in the training dataset. With 80:20 split between training and validation datasets we end up with 240 samples in the training dataset, each sample consisting of 8 forward prices, i.e., with 1,920 instances of forward price changes. This number can be compared with the number of weights that we aim to train. With 2 hidden layers and 10 activation units in each layer, we have 100 connections between activation units performing non-linear transformations. Thus, the number of weights is large enough to perform non-linear regression on complex curve shapes but small relative to the number of instances of forward price changes in the training dataset.

Figure 2 displays the curve moves predicted by ANN and PCA conditional on forward price at the 1M tenor, our control variable, moving by ±\$10/bbl. The PCA predictions are the transformations of the initial curve according to the shock suggested by the 1st principal component with the short end of the curve moving by ±\$10/bbl. Here, we keep in mind that the 1st principal component explains 94% of the variance and that all principal components are independent.

These two charts alone are sufficient to demonstrate the limitations of PCA and the richness of ANN. 1M forward price shocks applied to the flat \$80/bbl curve are completely in line with the picture predicted by PCA (1st principal component). But the same 1M forward price shocks applied to the flat \$20/bbl curve produce a very different picture with much tighter moves at the far end of the curve. This shows that ANN, unlike PCA, can clearly distinguish between the low and the high oil price regimes.

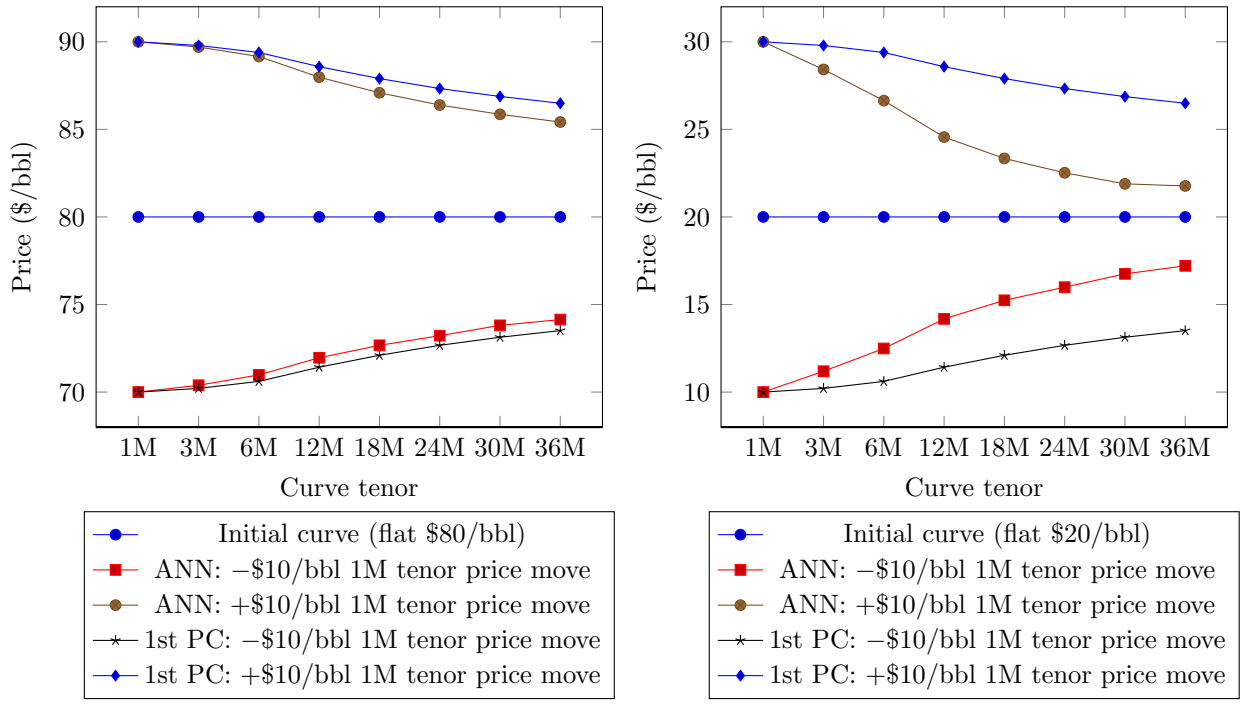

Figure 2: Flat curves ($80/bbl and $20/bbl levels)

Now, let us see what kind of dependency on the initial curve shape we can observe for the upward and downward sloping curves. Figure 3 displays ANN results for the relatively steep curves. The main result is a much stronger flattening of the upward sloping curve for the upward shocks than predicted by PCA, especially at the far end of the curve. When we start with a downward sloping curve, negative shocks (1M forward price moves down) make the curve flatter than predicted by PCA.

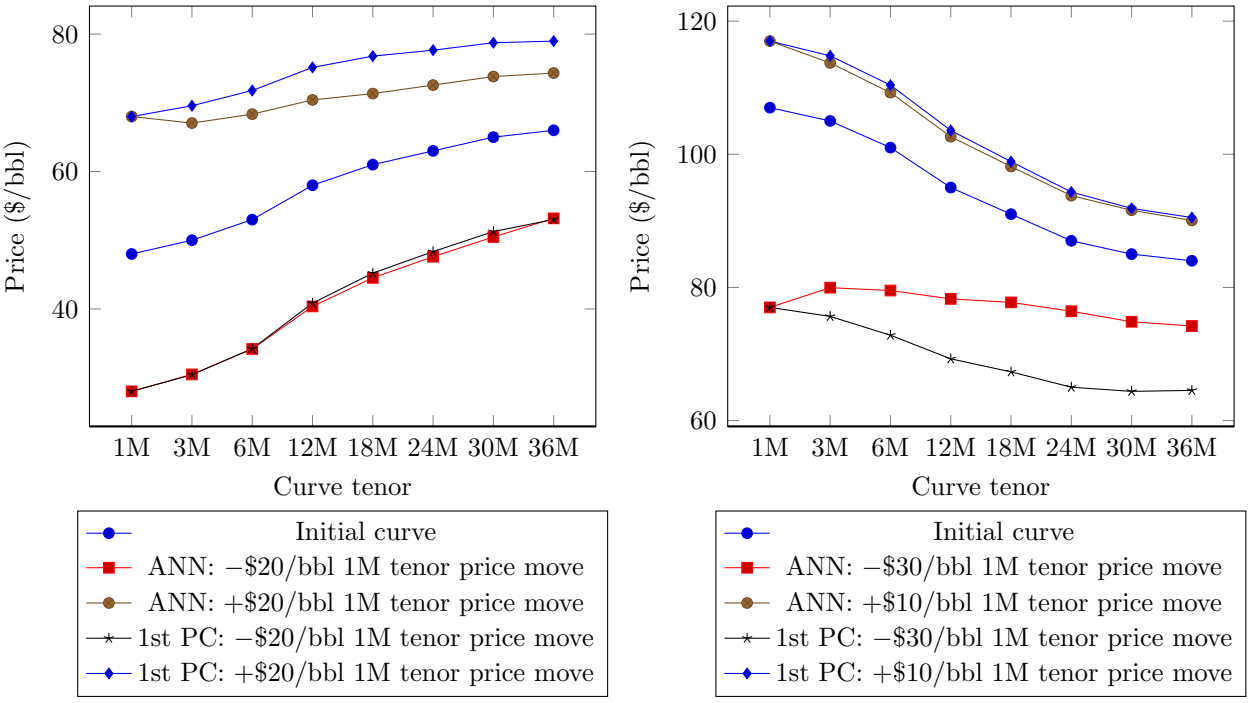

Figure 3: Upward and downward sloping curves

Interestingly, it is the curve behaviour at the far end of the curve that ensures strong curve flattening for the very different initial term structures considered above. In both cases ANN predicts a practically flat curve at the level of about \$80/bbl.

Now we can come back to Figure 2. We see a major, principal difference between ANN and PCA results. By construction, PCA can hold very limited information. Essentially, all information available to PCA is encoded in the covariance matrix of forward price returns at different curve tenors. Any information about the corresponding curve shapes is lost. But a large enough ANN with hundreds of trained weights associated with the connections between the activation units can store information about returns conditional on the shape of the curve.

However, the ability to extract and hold more information from the dataset comes with the danger of overfitting and poor generalization to the new, unseen data. It is natural to assume that a higher degree of regularisation would lead to a reduction in overfitting at the price of information loss. The objective should be to prevent massive overfitting without losing too much valuable information extracted from the dataset, i.e. not to swing to the other extreme of underfitting the ANN.

Figure (4) is a reconstruction of Figure 2 (flat \$20/bbl curve) with ANN results obtained for two values of the regularisation parameter: $\alpha = 0.05$ (strong regularisation) and $\alpha = 0.00005$ (original weak regularisation).

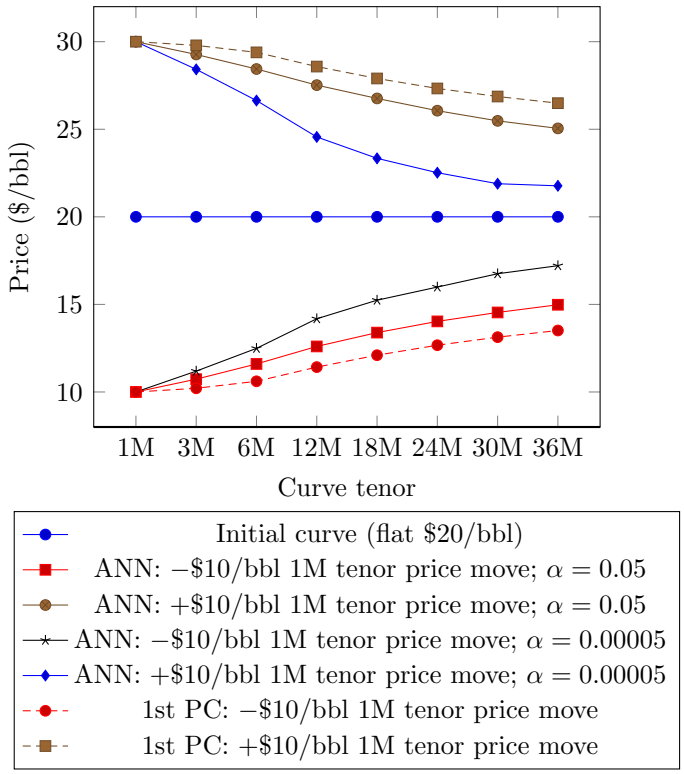

Figure 4: ANN regularisation

As expected, the larger value of the regularisation parameter leads to the convergence of the ANN-generated results to the ones obtained with PCA. Therefore, we can view PCA as a limiting case of ANN with strong regularisation where all term structure specific information is lost.

We can try to quantify by how much ANN outperforms PCA by predicting curve shape changes sensitive to the initial term structure. One possible way to do it is to compare predicted curve shapes conditional on price move at the 1M tenor produced by ANN and

the 1st principal component for samples from the validation dataset. The quantitative measure is the absolute error between predicted and actual realised forward prices at the given curve tenor. With 60 samples in the validation dataset and 8 curve tenors we have a total of 420 absolute errors since we match the 1M tenor forward price move by construction. The results are summarised in the following table:

Model	**Total absolute error ($/bbl)**	**Average absolute error ($/bbl)**	**Standard deviation ($/bbl)**
ANN	342.05	0.81	0.98
PCA	497.30	1.18	1.12

5.2 USD Swap Curve

The USD swap curve dataset consists of 250 monthly swap curve observations (June 1996 to April 2017) at 1Y, 2Y, 3Y, 5Y, 10Y, 15Y, 20Y and 30Y tenors. PCA conducted on the absolute monthly USD swap curve returns shows that the 1st principal component explains 86% of the variance and the 2nd principal component explains further 12% of the variance. Figure 5 plots eigenvectors for the first two principal components.

The 1st principal component eigenvector reaches maximum at 5Y tenor, indicating that the 5Y swap rate returns have the largest variance. The 2nd principal component eigenvector shows that curve pivots at the 5Y tenor. Therefore, 5Y swap rate is a natural choice for the swap rate that "drives" the whole curve, i.e., it is a natural choice for the feature that controls the magnitude and direction of the curve moves.

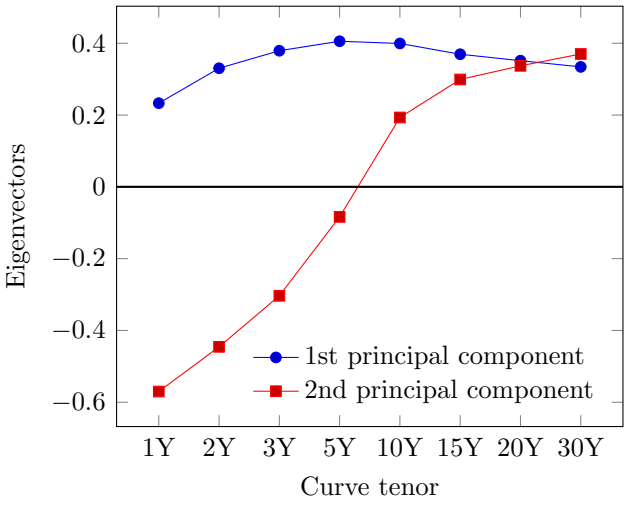

Figure 5: Principal Component Analysis for USD swap curve

The curve shapes obtained from different initial curves are fully consistent with the 1st principal component even for the small value of regularisation parameter ($\alpha = 0.00005$) as we can see in Figure 6. However, the curve shapes are not totally symmetrical with respect to up and down moves and there are minor but noticeable deviations from curve shapes suggested by the 1st principal component.

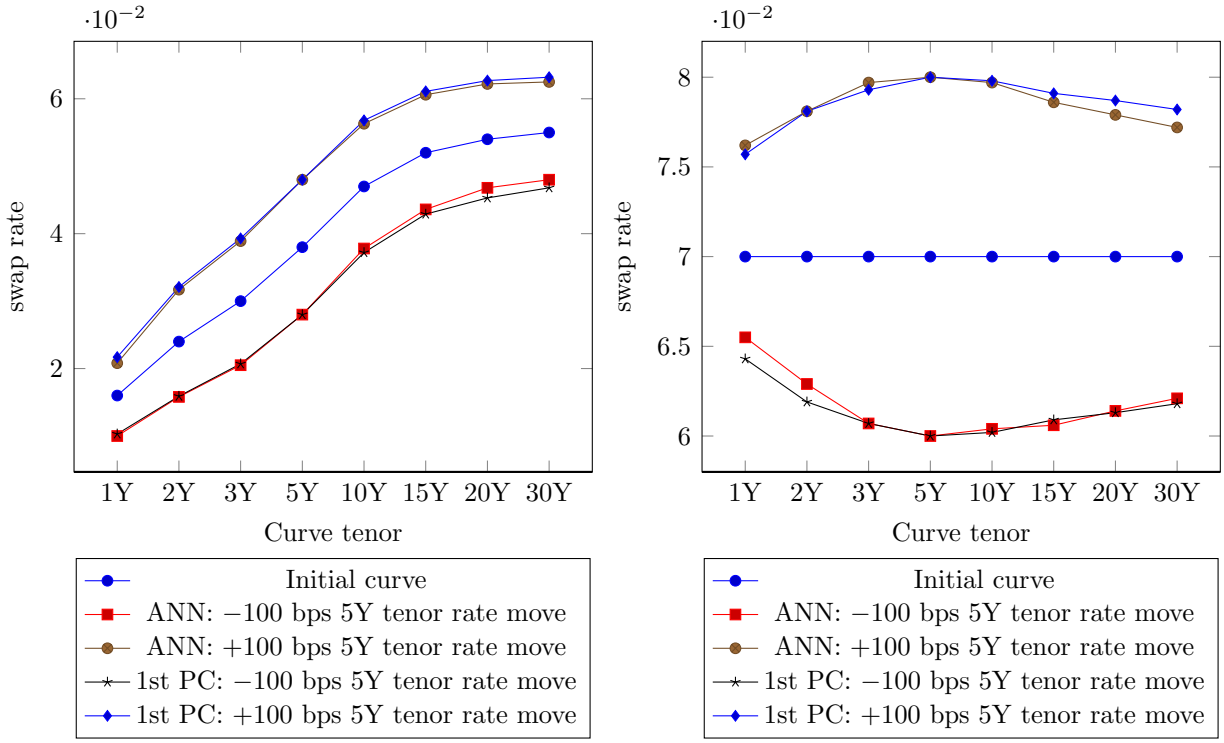

Figure 6: Swap curve moves for different initial term structures

As we have already seen in the case of brent crude oil forward prices, properly trained ANN is able to distinguish between different curve shapes and would suggest different dynamics to different initial term structures. These differences are not particularly large for the USD swap curve due to the fact that the curve moves in the training dataset belong to the same regime (e.g., no pronounced curve inversions, etc).

Nevertheless, these second order effects can be captured by ANN and can be used for predicting more exact shapes of the curve in various curve flattening and steepening scenarios. The following table shows dependence of the curve moves at the short end (1Y swap rate) on the initial term structure in response to the curve moves at the 5Y tenor.

Scenario	**5Y tenor move**	**1Y tenor move**	**1Y tenor move difference**
Upward sloping curve	+100 bps	+48 bps	+14 bps
Flat curve	+100 bps	+62 bps	
Upward sloping curve	−100 bps	−60 bps	+15 bps
Flat curve	−100 bps	−45 bps	

PCA would predict the same curve moves at 1Y tenor regardless of the initial term structure.

5.3 Long-Term Moves

So far we have performed statistical analysis of the curve transformations over a relatively short time interval: 1 month. Would the current shape of the curve be relevant if we want to predict possible curve transformations over much longer time horizons? To answer this question we have run the ANN, specified in the previous sections, on 5-year absolute brent forward price returns with $\alpha = 0.00005$, the low regularisation level. Figure 7 shows that

the initial shape of the curve plays a progressively smaller role as we extend the prediction time horizon when we are interested in the relative moves of the forward prices at different tenors.

This is an important qualification: the ANN constructed according to the rules outlined in Sections 2 and 4 is designed to predict the curve shape transformations conditional on the changes in value of the control variable − for our dataset we have chosen this variable to be the 1M forward price.

An alternative approach would be to predict the curve shapes directly, i.e. not conditionally on the value of control variable. This can be a productive approach when we study long term curve transformations. Fortunately, our framework allows us to do this with only minor changes to the ANN design. All we need to do is to reduce the number of inputs by 1 (remove the control variable) and make changes to the output vector. Equations 4 and 5 now become:

$$InputVector(t) = [P_{T_1}(t), \ldots, P_{T_K}(t)] \tag{6}$$

$$OutputVector(t) = [P_{T_1}(t + \Delta t), \ldots, P_{T_K}(t + \Delta t)] \tag{7}$$

with Δt now set equal to 5 years rather than 1 month.

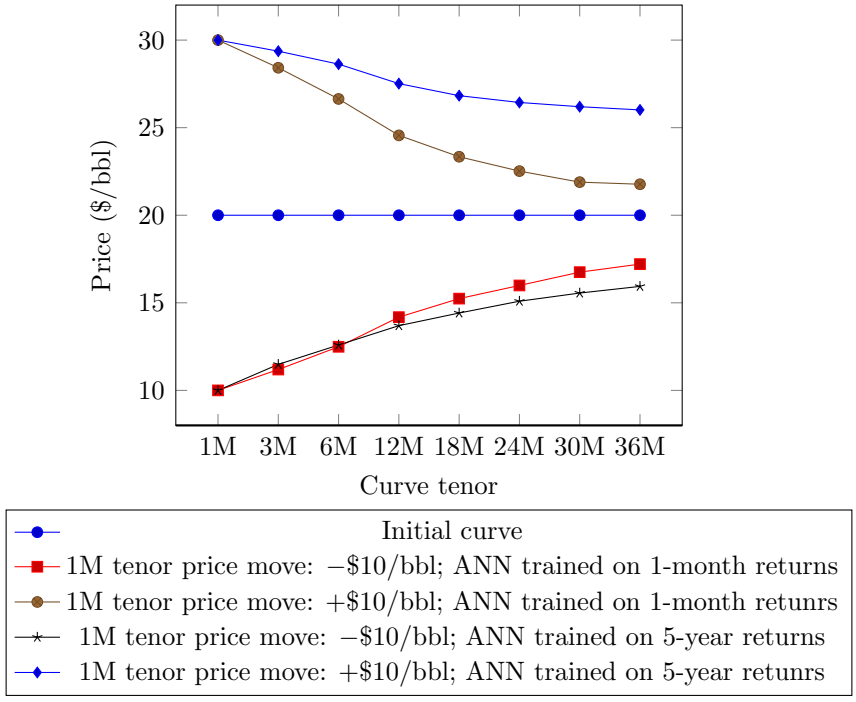

Figure 7: ANN trained on short-term and long-term curve moves

It looks like the result is a picture of long-term convergence to some average price level (Figure 8). Perhaps unsurprisingly, ANN tells us that if we start at a historically low level then it is likely that in 5 years' time the forward curve will be higher and if we start at a historically elevated level then in 5 years' time we can expect the curve to be lower.

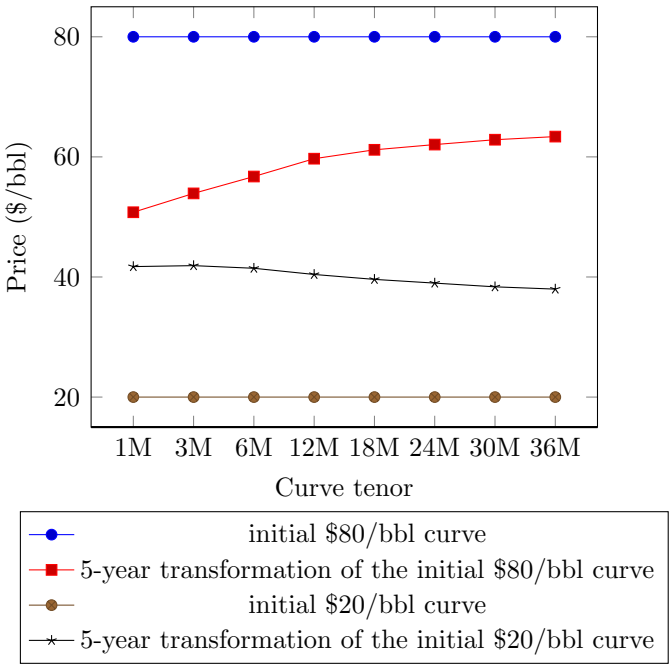

Figure 8: ANN trained on long-term curve transformations

Figure 8 indicates that the long-term average level is somewhere between $40/bbl and $60/bbl. This statement seems to be almost trivial and illustrates the loss of the term structure information as Δt increases. It is also worth noting that curve transformations are consistent with the PCA results (larger variance at the short end of the curve).

5.4 Autoencoders

An autoencoder is an ANN learning routine which trains the network to replicate the input via a bottleneck structure [9]. The number of activation units in hidden layers should be less than the number of inputs. This creates a cost effective representation of inputs. In other words, the autoencoder tries to find a low-dimensional representation (a simpler factor model) of the input data.

Therefore, following our notations, an autoencoder can be constructed as the following feedforward MLP network:

MLP Parameter	**Value**
Number of units in the input layer	Number of curve tenors
Number of units in the output layer	Number of curve tenors
Number of hidden layers	2
Number of units in each hidden layer	`int`(Number of curve tenors / 2)
Type of activation function	Hyperbolic tangent

with equations 4 and 5 now replaced by

$$InputVector(t) = OutputVector(t) = [P_{T_1}(t), \ldots, P_{T_K}(t)] \tag{8}$$

Due to the fact that an autoencoder, by construction, is only able to learn the key features of the data structure and ignores the secondary effects, it can be viewed as a model that

performs smoothing of the input curve. The "natural" curve shapes, which can be defined as sufficiently smooth, should be replicated by the autoencoder almost exactly. At the same time curves that exhibit spikes or steep gradient changes are likely to be replicated with errors. If we define replication error as a difference between the original price (rate) at the given tenor and the price (rate) predicted by the autoencoder then we have a readily available quantitative measure that can help us to detect abnormalities in the input term structures.

Figure 9 shows curve replication with an autoencoder trained on the brent dataset with the regularisation parameter set at $\alpha = 0.00005$. The left chart is almost exact replication of the term structure from the training dataset, even though an autoencoder only had four activation units in each of the two hidden layers. The right chart shows result of running the same autoencoder on the distorted term structure with a sharp bend at the 1Y tenor and a flat term structure after that tenor.

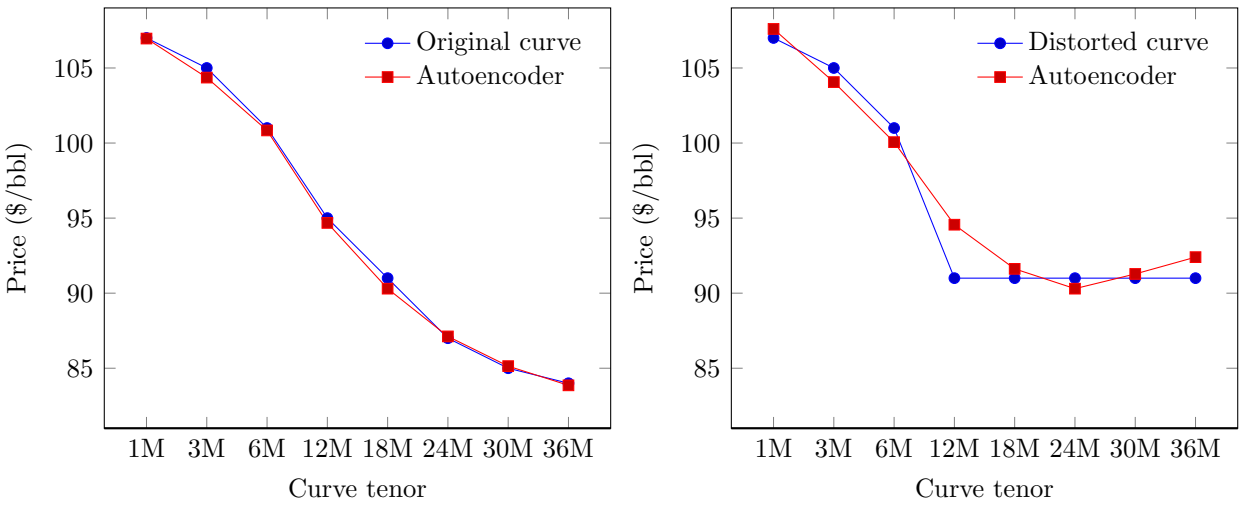

Figure 9: Autoencoder

The sum of absolute errors across all tenors in the former case is \$2.26/bbl while it is \$9.04/bbl in the latter. This large difference in the magnitude of errors is a strong indication that the distorted curve is in some way unnatural. The autoencoder also suggests a normalisation of the curve with the forward price at the 12M tenor moving up by \$3-4/bbl. This can be viewed as meaningful signal.

5.5 Parametric Model vs. ANN

Now is the time to ask the following question: how much information about the curve moves condtional on the initial curve shape can a standard parametric model preserve? After all, ANN itself is just a non-linear multi-factor model. Can a parametric model calibrated on the given training dataset preserve the covariance structure as a minimum and reproduce rich term structure dynamics as a maximum?

It appears that the answer to this question is both yes and no. Yes, properly calibrated parametric model can reproduce PCA results. And no, it can hardly move beyond PCA in terms of distinguishing between different initial term structures.

As an illustrative example we take the Ornstein-Uhlenbeck process, a stationary mean-reverting Gauss-Markov process, which is calibrated through regression on the same brent crude oil dataset we used to train our ANN. The Ornstein-Uhlenbeck process is specified in the form of the following stochastic differential equation:

$$dS(t) = \lambda\left(\mu - S(t)\right)dt + \sigma dW(t)\,, \tag{9}$$

where $S(t)$ is the variable we want to model (either an interest rate or a forward commodity price), $W(t)$ is the driving Wiener process, σ is the volatility, and μ is the level of mean reversion. This equation in the form of a 1-factor short-rate model (Hull-White, extended Vasicek) is a popular choice for the modelling of the interest rate dynamics. We recommend [10] for the latest research in this area.

Solution to the Ornstein-Uhlenbeck SDE has the following form:

$$S(t+\delta) = S(t)e^{-\lambda\delta} + \mu\left(1 - e^{-\lambda\delta}\right) + \sigma\sqrt{\frac{1 - e^{-2\lambda\delta}}{2\lambda}}N(0,1)\,. \tag{10}$$

Given large enough dataset of historical observations, the calibration can be done using regression (e.g., by minimising the sum of squared errors):

$$S(t+\delta) = aS(t) + b + \varepsilon$$

where δ is a time step and

$$a = e^{-\lambda\delta}\,,$$
$$b = \mu\left(1 - e^{-\lambda\delta}\right)\,,$$
$$\mathrm{stderr}(\varepsilon) = \sigma\sqrt{\frac{1 - e^{-2\lambda\delta}}{2\lambda}}\,.$$

Then the model parameters are given by the following expressions:

$$\lambda = -\frac{\ln a}{\delta}\,, \qquad \sigma = \mathrm{stderr}(\varepsilon)\sqrt{\frac{-2\ln a}{\delta(1-a^2)}}\,.$$

Reggression is performed for each curve tenor separately. Parameters λ and σ are determined for each T-month forward price (T = 1M, ..., 36M). The new forward prices (the new simulated term structures) are generated by applying the δ-period shocks to each term structure in the original dataset. With δ being set equal to 1 month we get 1-month curve transformations for all terms structures observed in the original dataset (300 samples). The shocks to the curve (as per equation 10) are driven by the standard normal random variables – a single standard normal random variable is generated for the given sample, i.e. forward prices for all curve tenors are driven by the same random variable. But these standard normal random variables are independent across samples (independent shocks are applied to each term structure in the dataset).

The first test is to see whether the forward price returns generated using the Ornstein-Uhlenbeck process would produce the same PCA results we obtained for the original dataset. The calculations show that this is indeed the case with the 1st principal component capturing 97% of the variance and the 2nd principal component capturing 2% of the variance. This is close enough to the 94%/5% split we see for the original dataset. The shapes of the corresponding eigenvectors are practically identical for both the original and the generated datasets. The second test is comparison of scenarios (curve moves) produced by the ANN and the parametric model for the same intial term structure. The control variable is 1M forward price move which is matched by construction. Figure 10

displays the parametric model results for the downward sloping curve against the ANN results we saw in Figure 3.

The parametric model scenarios are very close to the PCA but fail to generate the same degree of curve flattening as the ANN. The ANN suggests that the $30/bbl downward move at the short end of the curve corresponds to the curve flattening scenario with the end result being practically flat curve with zero gradient between 1M and 36M tenors. The curve is predicted to become flat at the $80/bbl level. In contrast, the parametric model predicts only moderate flattening of the curve for the −$30/bbl short end move scenario. The resulting gradient between 1M and 36M tenors is −20%.

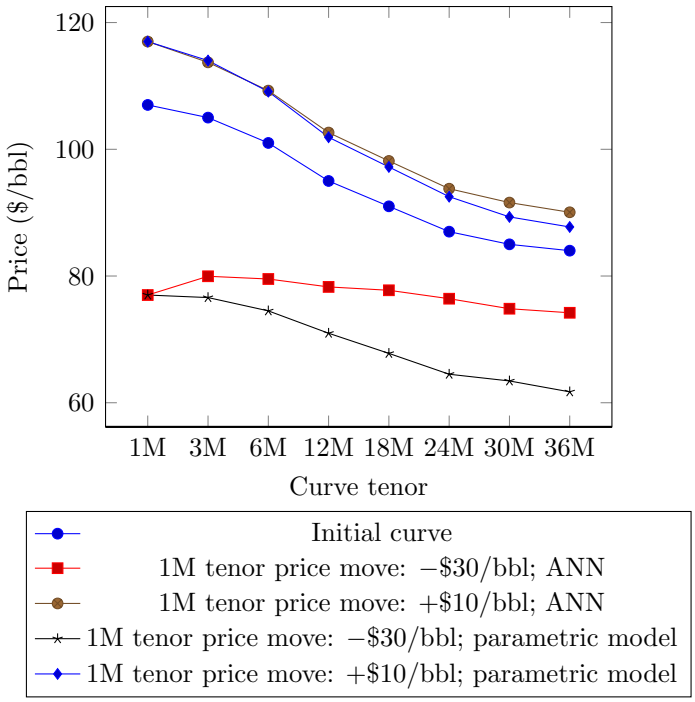

Figure 10: Downward sloping curve - ANN and parametric model

6 Conclusion

The ANN approach has potential to replace PCA as the primary non-parametric curve analysis tool. ANN can match PCA results in the limit case of strong regularisation. At the same time, ANN has capacity to store more information about the dataset. This gives us a possibility to find the right balance between under- and overfitting according to the specific problem we have to solve.

ANN configured and trained as an autoencoder is an example of a strongly regularised network designed to learn the low-dimensional representation of the dataset. It can be used to detect deviations from the normal smooth curve shapes and indicate directions in which the curve is likely to be rectified.

Acknowledgements

The author would like to thank Dr. Alexander Sokol for many inspiring discussions. His insights and assistance in writing this paper are very much appreciated.

Disclaimer and Declaration of Interest

The author reports no conflict of interest. The author alone is responsible for the content and writing of the paper. The opinions expressed are those of the author and do not necessarily reflect the views and policies of Standard Chartered Bank. All figures are based on the own calculations.

This paper is for information and discussion purposes only and does not constitute either an offer to sell or the solicitation of the offer to buy any security or any financial instrument or enter into any transaction or recommendation to acquire or dispose of any investment.

References

[1] G. Cybenko, Approximation by Superpositions of a Sigmoidal Function, *Math. Control Signals Systems*, **2** (1989), 303-314.

[2] K. Hornik, M. Stinchcombe, and H. White, Universal Approximation of an Unknown Mapping and Its Derivatives Using Multilayer Feedforward Networks, *Neural Networks*, **3** (1990), 551-560.

[3] H. Amilon, A neural network versus Black–Scholes: a comparison of pricing and hedging performances, *Journal of Forecasting*, **22**, Issue 4 (2003), 317-335.

[4] S. Raschka, *Python Machine Learning*, Pact Publishing (2015).

[5] C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer (2006).

[6] C. Baumeister, P. Guérin, and L. Kilian, Do high-frequency financial data help forecast oil prices? The MIDAS touch at work, *International Journal of Forecasting*, **31** (2015), 238-252.

[7] N. S. Grønborg and A. Lunde, Analyzing Oil Futures with a Dynamic Nelson-Siegel Model, *Journal of Futures Markets*, **36** (2016), 153-173.

[8] J. Shlens, A Tutorial on Principal Component Analysis, Working Paper, arXiv:1404.1100 (2014).

[9] J. B. Heaton, N. G. Polson, and J. H. White, Deep Learning for Finance: Deep Portfolios, Working Paper, SSRN 2838013 (2016).

[10] J. Hull, A. Sokol, and A. White, Short-rate joint-measure models, *Risk*, October (2014).