A Survey on Machine Learning Applications In Finance

Alexander Swanepoel[1]

[1]**University of Warwick, Computer Science MSc**

Machine learning possesses properties of inference and prediction on the basis of known quantities, allowing application towards finance inexorable. The utilisation of neural networks, and data mining has been pioneered in finance since the 1990s. Whilst thousands of papers have been published on this subset of topics, the emergence of the technical advantages remain somewhat unmapped, and substantial research is sparse. We seek to propose a comprehensive introduction to nonparametric methods.

Keywords: Machine Learning, Finance Research, Data Mining, Option Pricing

INTRODUCTION

Let us consider the informal skeleton of machine learning, linear regression; in essence, input-output mapping. An artificial neural network (ANN) can be considered to be a superposition of functions, producing a similar mapping result. By implementation of a loss function, one is able to optimise the ANN, analogous to choosing regression components such that the function mapping is optimal. One may choose to design the loss function such that the Manhattan distance and the output is resolved to achieve this. With this prerequisite condition met, one is able to understand how via assertion of the universal approximation theorem, the approximation of the ANN functions are suitable, and accurate with respect to the appropriate conditions. In options pricing, the access of these functions may be paramount to success. For example, suppose that you have access to a sufficiently large dataset containing information of implied volatility, vega and delta ratios, hedging ratios or data of a similar sort. The result of an ANN function could be directly compared to historical options prices, and if mirrored with a reasonable accuracy, implemented.

The term machine learning can be considered to be difficult to define formally, due to the lexeme being an umbrella term. Similarly to artificial intelligence, where machine learning is a subset of this. Let us form a pragmatic definition of machine learning for understanding purposes, as the definition could be debated for a substantial amount of time. The definition we seek to imply when we're invoking machine learning, may overlap with statistical learning. Machine learning can be said to be the application of regression analysis algorithms to specific criterion such as minimising estimators between the predicted outputs and real outputs. Within machine learning are the elements unsupervised and supervised learning. Unsupervised learning as implied requires no supervision of the model, and often is applied to unlabelled data. Supervised learning will act upon the result of the previous iteration of the model, improving the model incrementally towards optimisation. Unsupervised learning will undergo analysis, clustering dependencies of variables in the data, and by proxy reducing a multi-dimensional dataset, to a potentially smaller set of dimensions, where the variables that have a negligible impact on the output can be ignored, hence known as "dimentionality reduction"; however note the clustering of variables is not ideal, as this could be considered more pioneering. More recent mechanisms would include neural networks and matrix factorisation. Supervised learning would consist of input-output mapping as previously described, $f : X \rightarrow Y$, where $f \subset X \times Y$, containing pairs $(x, f(x)) \ \forall \ x \in X$. Usually, inference of Y can be measured via input parameters of X by development of an estimator, by minimising the deviations in the estimator you are able to achieve optimal results. For example, many neural network models used in finance would historically use a Mean Squared Error estimator, and map deviations of the predicted model, to deviations of a benchmark model such as the Black-Scholes[1].

The simplest mathematical formulation of the model given as,

$$C(s,t) = N(d_1)S_t - N(d_2)PV(K)$$
$$d_1 = \frac{1}{\sigma\sqrt{T-t}}\left[ln\frac{S_t}{K} + (r + \frac{\sigma^2}{2})(T-t)\right]$$
$$d_2 = d_1 - \sigma\sqrt{T-t} \tag{1}$$

There are many variants of the Black-Scholes to include parameters such as implied volatility, calibrated volatility, etc. and will be specified as such when used as a benchmark. The major estimators concerned, will be the following:

Estimator List	
Estimator Name	Formulation
Mean Squared Error	$\frac{1}{N}\sum(\hat{y}_i - y_i)^2$
Mean Absolute Error	$\frac{1}{N}\sum\lvert\hat{y}_i - y_i\rvert$
Mean Absolute Percentage Error	$\frac{1}{N}\sum\frac{\lvert\hat{y}_i - y_i\rvert}{y_i}$
Mean Percentage Error	$\frac{1}{N}\sum\frac{\hat{y}_i - y_i}{y_i}$
ExpShort*[14]	$-\frac{1}{\alpha}\int_0^{\alpha} VAR_{\gamma}(X)d\gamma$

*Novel estimator known as expected shortfall, implemented in H. Buehler et. al(2019) [2]; where $X \in L^{(p)}\mathscr{F}$ is portfolio payoff in a time period $0 < \alpha < 1$; VAR_{γ} is value at risk.

Similarly to the Black-Scholes, other comparators can be used to determine the output(Y) to an accurate approximation; A list of the most frequently used learning algorithms in machine learning may include:

- Hull-White Model (2017)[8] - Approximation of delta function as a quadratic function in Black-Scholes Model to yield a ’minimum variance delta’(Hull-White)
- Non-Parametric-Kernel Regression (NPKR)
- Longstaff-Schwartz (2001)[11] - Approximation of American-Style options using least square methods, and jump diffusion application (Long-Sh)
- Support Vector Machines

- Linear Regression Analysis
- Carr et. al Model (2003) - Approximation of options pricing using stochastic volalility jumps (Carr)
- Black-Scholes Generalised to Local Volatility [5] - Or, Black-Scholes Local Volatility (BSLV)

The origins of machine learning in finance, commenced with artificial neural network back propagation, sought promise in arbitrage opportunities and technical analysis, yet was deemed lacking [7]. Followed closely by M. Malliaris and L. Salchenberger (1993) [12]; applying a single layer neural network on options pricing. The study used input features subject to stock and strike price, time to maturity, interest rates and at-the-money implied volatility, and provided comparison to the Black-Scholes model for benchmark analysis. The dataset operated upon was the S&P100 6-Month, which although was an appropriate and realistic application of the ANN, would not be implementable today as the index metrics were shifted in 1993 by the Chicago Board Options Exchange Market Volatility Index (VIX); then altered in 2003 to have the same basis as the S&P500. Whilst the model performed relatively well with respect to estimation, it consistently underperforms with in-the-money inference. Under the performance metric of mean absolute error, it outperforms the Black-Scholes 50 percent of the time.

Following, a substantial contribution toward the research field includes Dugas et al. (2009) [4] via creation of a free-arbitrage principle condition model, where a shift in the pricing of an underlying option does not shift the output proportionally, however has dependency on higher-order terms on the modelling function. The output of the model is given as a ratio of the strike price with respect to the underlying option price, i.e. 'moneyness' with input metrics such as strike-stock price ratio, and the time to maturity. However, the paper assumes no use of organic data, and uses artificial datasets of the S&P500 5-year. Even though this may be sufficient for theoretical approximations, organic data would be ideal to test real-world implementation. The research also utilises a multi-layer perceptron as a benchmark whilst implementing the class function in standard artificial neural networks, and is able to consistently outperform.

MODERN APPLICATIONS OF MACHINE LEARNING IN THE LITERATURE

As of late, there have been numerous examples of intricate application of machine learning systems into options pricing and hedging. The technologies presented have been greatly complex relative to their pioneer counterparts, and have used complex techniques such as deep learning. Deep learning is an extrapolation of artificial neural networks, with the excemption that they contain multiple layers; progression through the layers accesses higher-level information. For example, a deep learning network applied to recognition of a painting, may include on the first layer, information about the edges and vertices of the application space, to move onto the next layer, contain colours, shades and hues, to progress through the layers to more complex information such as brush inflections and painting style.

A recent paper published in 2018 by Ferguson and Green [6], in which deep learning was used to model European-style options pricing, using a six hidden-layer deep learning network. The input parameters used were stock price, time to maturity, implied volatility, and correlations between underlying options prices. Whilst impeccably accurate, the model was conducted on the basis of a basket option, containing 6 options in total, and considered the worst-of-basket option. Whilst this is plausible, basket options are predominantly only available to institutional investors; this has the implication that the deep learning algorithm may be of lesser application to a retail investor.The output(Y) consisted of the option price, and was evaluated relative to a Monte Carlo simulation with respect to Brownian motion; finding the most success the deep neural network relative to the 500M dataset, at a 10K Monte Carlo Path simulation. Whilst this is a significant result, the difference to 50M dataset, at 100K Monte Carlo paths, is negligible and the computational requirements are extensive for the prior, requiring 1.5 petabytes of disk space, which is slightly unfeasible. The remarkable features of the paper include the accuracy of the mean squared error coefficients yielding consistently 0.999-0.997 values, as well as the sheer speed in which the computational processes delivered. Realistically speaking, the feature of value is prediction time, in which the most valuable quality would be how quickly are you able to produce a simulated option price with respect to the training of the model on a given dataset. The results they found were somewhat unsurprising, concluding that the speed of valuation has dependency on the size of a given dataset, however, they were able to train at the upper ranges a model consisting on six hidden-layer neuron networks, at 1400 nodes per layer, producing 50K valuations in under 6 milliseconds; this is a truly remarkable result.

Thus far, we have only considered options pricing applications, however the depth of machine learning application in finance is broad and diverse by nature of the technology. For example, Cheng & Cirillo [3] satisfying huge demand for a model capable of accurately depicting and modelling defaulted recovery rates and times. The sub field of which this finds application can be considered to be credit risk management. Even though machine learning was not directly used to produce the model itself, Bayesian updating was used to train the model to an acceptable accuracy using prior-posterior distribution techniques. Whilst the model is excellent and satisfies a well defined problem in the field, it is noteworthy that as with most Bayesian probability models, prior knowledge of the case is required.

A novel and creative use of machine learning in the field of finance, is the application of deep learning, to produce an image recognition algorithm to analyse technical indicators in stock chart patterns for inference, performed by Sezer & Ozbayoglu [16]. In essence, a convolutional neural network was used to take inputs as 2D Matrix arrays, the 2D arrays consisted of candlestick datasets with daily open and close prices, this was then massaged and labelled, and ran through a nine-layer network of complex processing, using drop-out layers to handle overfitting. The input parameter(X) considered over a span of 15 different technical indicators produced a remarkable profit with respect to the Buy-and-Hold strategy (popularised by Warren Buffet) as commonly used as a benchmark. The algorithm implemented for modelling remains elegant and easy to implement, utilising *Keras* and *Tensorflow* modules. Whilst I am unable to comment on the efficacy of the techniques used relative to other convolutional neural network techniques available, I am able to recognise that although a broad range of technical analysis may be a significant sample, the validity of the indicators themselves may not hold in modern circumstances, for instance the weighted moving average and crossover methods described in the paper itself have been known since the early 1990s; this poses the question as to whether or not they will produce a significant enough advantage in the market to yield profit. With this in account, the paper itself is a creative implementation of machine learning not previously seen, and may potentially create a sub field of financial image processing with respect to technical analysis, and hold as a pioneering piece of research in an emergent field.

Buehler et. al(2019)[2] present an elegant formulation of a generalised algorithm, giving life to the data-driven approach of options pricing, to account for real-world variables such as transaction costs, market impact and friction, and other various constraints. The necessity for the research itself was well recognised as the algorithm does not depend on specific market dynamics, a noteworthy feature, with respect to computation is the algorithm does not depend on portfolio size, as previously denoted by Ferguson et al. [6]; contesting yet another specification. Buehler et. al consider hedging strategies with input parameters(X) as the logarithm of the stock price $log(S)$; they also consider more adventurous and extravagant options, however due to my unfamiliarity I cease to comment on the latter. The output parameter (Y) considered is the hedging ratio, and have used the Black-Scholes with implied volatility as a comparator. They were also the first to implement the expected shortfall estimator into their model, allowing to consider niche extreme cases of loss, at the tail end of the distribution; this was a huge implementation in the field, as hedging ratios were often skewed by outlier events that had not taken into account the devastation of an extreme asymptotic loss. Although the model performs exceptionally well in relatively normal dimensionality cases, the performance does not hold for higher-dimensionality, but has produced research to highlight the potential of their algorithm when applied to high-dimensionality cases. The algorithm itself, essentially replicates the mind of a successful trader, accounting for realistic measurements, however being able to exclude fundamental analysis, bias, or emotion. The algorithm itself encodes for a minimisation in variance of the portfolio by comparison of profit-loss distributions, expressing it as a function, and optimising the function itself. The research itself poses an opportunity for further higher-dimensional algorithm construction with respect to deep learning.

Spurring on the data-driven assessment of options, is Liu et. al(2019)[10], applying artificial neural networks to European-style options pricing, ignoring the problem of higher dimensionality, and specifying the approach to solve nonlinear partial differentials. The model itself is under comparison to the Black-Scholes, emerging as an industry benchmark. The input parameters would consider the ratio of the strike price with respect to the underlying option price i.e. 'moneyness' and the time to maturity,producing the output parameter of implied volatility, a unique output considering the previous examples. Another peculiarity of the research is the plethora of estimators used as a performance metric; the estimators themselves span the mean absolute error, the mean absolute percentage error, and the mean squared error. The performance was exceptional across all margins.

Palmer (2019)[13] proposes a new and untried implementation of ANN in the form of an evolutionary algorithm directed at options hedging. Please note the implementation was conducted as partial fulfilment of a PhD thesis, and has not been microscopically examined, however does not remove the influence and impact of the work none the less. The paper takes into account the moneyness, implied volatility as a product of the root of the time to maturity, and interest rates, and exercises them in an evolutionary algorithm known as the particle-swarm optimisation, where Palmer shifted the algorithm to account for crossover and mutation. This is in my opinion the most creative algorithm implemented in options pricing ANN to date.The model itself uses estimators of mean absolute error and mean absolute percentage error. The output parameter is the ratio of the option price with respect to the strike price(moneyness); with relative benchmarks of the Black-Scholes and LongSch Model as denoted previously. Once again, another pioneering strategy enveloped, the accounting of a modern benchmark model, only implemented by one other previously[9] ensuring realistic application.

Lastly, we consider the most recent literature in the field to date; Ruf & Wang [15]. This paper in my opinion is the most comprehensive application of ANN to date, with real-world implementation. The algorithm itself, designed to produce end-of-day tick options pricing, consists of a cornucopia of input parameters range from moneyness, to implied volatility as the product of the root of the time to maturity, delta(Δ)-vega(ν) functions, and even the second-order derivative of the delta as in input, known as Vanna. This is the most state-of-the-art algorithm to date in my opinion. They use the industry standard benchmarks, Black-Scholes accounting for implied volatility, as well as a linear regression statistical analysis, whilst also using the Hull-White benchmark for additional measures, and a mean square error as an estimator. The full computational algorithm has a substantial improvement in hedging relative to the Black-Scholes, however was unable to infer the non-linear characteristics. A superposition of greeks as well as regression analysis is still a viable candidate for the capture of non-linearity. They also made note of a remarkable repercussion of data cleaning, resulting in leakage that in itself has been quantified to show a tangible loss. This is in itself a new consideration to be accounted for in the sub field, and leads to potential future research, posing the question, to what magnitude is the loss due to data leakage?

DISCUSSION

A summation of all the characteristics and parameters for options described above can be found here:

Summary				
Research	(X)	(Y)	Relative Bench	Estimators
Ferguson[6]	Stock-price, time to maturity, Implied volatility and underlying correlations	Option Price	Monte Carlo	Mean Squared Error
Liu et. al [10]	Moneyness, time to maturity	Implied volatility	Black-Scholes	Mean absolute error, Mean absolute percentage error, mean squared error
Palmer [13]	Moneyness, implied volatility*sqrt(time to maturity), interest rates	ratio of Option price and Strike price	Black-Scholes-implied-vol, LongSch	Mean absolute error, Mean absolute percentage error
Buehler et al. [2]	log(StrikePrice)	Hedging Ratio	Black-Scholes	Expected Shortfall
Ruf & Wang	Moneyness, implied volatility*sqrt(time to maturity), delta-vega functions	Hedging Ratio	Black-Scholes-implied-vol, Hull-White, Regression	Mean Squared Error

Comments

Whilst machine learning has seen substantial application in finance, the range of topics and creativity of algorithms is sparse. This may be due to the complexity of the mathematics involved when merging the two fields of economics and computer science, however the product of such an endeavour is well rewarded as the operation space remains relatively unexplored. The sub field itself seems in its adolescence and has a set of emergent research questions posed. Due to the sparseness of the literature, the majority of neural networks in finance and economics have been primarily focused around hedging and options trading, due to their intrinsic nature to be excellent approximators of non-linear equations such as partial differentials. Within the research itself, it seems as vast undiscovered application for neural networks and machine learning algorithms.

REFERENCES

[1] F. Black and M. Scholes. The pricing of options and corporate liabilities. *Journal of Political Economy*, 81(3):637–654, 1973.

[2] H. Buehler, L. Gonon, J. Teichmann, and B. Wood. Deep hedging. *Quantitative Finance*, 19(8):1271–1291, 2019.

[3] D. Cheng and P. Cirillo. A reinforced urn process modeling of recovery rates and recovery times. *Journal of Banking Finance*, 96:1 – 17, 2018.

[4] C. Dugas, Y. Bengio, F. Bélisle, C. Nadeau, and R. Garcia. Incorporating functional knowledge in neural networks. *Journal of Machine Learning Research*, 10:1239–1262, 06 2009.

[5] B. Dupire, T. B. M. (see Black, and G. Options. Pricing with a smile. *Risk Magazine*, pages 18–20, 1994.
[6] R. Ferguson and A. Green. Deeply learning derivatives. *High Performance Computing eJournal*, 2018.
[7] D. D. Hawley, J. D. Johnson, and D. Raina. Artificial neural systems: A new tool for financial decision-making. *Financial Analysts Journal*, 46(6):63–72, 1990.
[8] J. Hull and A. White. Optimal delta hedging for options. *Journal of Banking Finance*, 82:180–190, 2017.
[9] H. Jang and J. Lee. Generative bayesian neural network model for risk-neutral pricing of american index options. *Quantitative Finance*, 19(4):587–603, 2019.
[10] S. Liu, C. Oosterlee, and S. Bohte. Pricing options and computing implied volatilities using neural networks. *Risks*, 7(1):16, Feb 2019.
[11] F. Longstaff and E. S. Schwartz. Valuing american options by simulation: A simple least-squares approach. *Review of Financial Studies*, 14(1):113–47, 2001.
[12] M. Malliaris and L. Salchenberger. A neural network model for estimating option prices. *Appl. Intell.*, 3:193–206, 09 1993.
[13] S. Palmer. Evolutionary algorithms and computational methods for derivatives pricing. 02 2019.
[14] R. T. Rockafellar, S. Uryasev, et al. Optimization of conditional value-at-risk. *Journal of risk*, 2:21–42, 2000.
[15] J. Ruf and W. Wang. Hedging with neural networks, 2020.
[16] O. B. Sezer and A. M. Ozbayoglu. Algorithmic financial trading with deep convolutional neural networks: Time series to image conversion approach. *Applied Soft Computing*, 70:525 – 538, 2018.