Neural Networks for extracting implied information from American options

Shuaiqiang Liu, Álvaro Leitao, Anastasia Borovykh and Cornelis W. Oosterlee

Motivation and proposal

citic

- Extract implied information from observed option prices with early-exercise features.
- Computing American-style option prices is generally more challenging than pricing European-style options.
- The optimization process to address the inverse problem requires solving the pricing model many thousands of times.
- Other complicating factors: negative interest rate and/or dividend yields (complex-shaped early-exercise regions)
- Implied volatility: inverse function approximated by an artificial neural network.
- Implied dividend: the inverse problem formulated as a calibration problem.

Outline

citic

1. Problem formulation

2. Artificial Neural Networks
   2.1 ANN for implied volatility
   2.2 ANN for implied dividend (and implied volatility)

3. Numerical results

4. Conclusions

Problem formulation

- For simplicity, we consider the Geometric Brownian Motion (GBM) process,

$$dS_t = (r - q)S_t \mathrm{d}t + \sigma S_t \mathrm{d}W_t,$$

  where S_t is the underlying price, σ is the volatility.

- The arbitrage-free value of an American option at t is

$$V_{am}(S_t, t) = \sup_{u \in [0,T]} \mathbb{E}_t^{\mathbb{Q}}[e^{-r(T-t)} H(K, S_u) | S_u],$$

  where $H(\cdot)$ is the payoff function with strike price K and expiration time is T.

Problem formulation

- An optimal exercise boundary $S_t^* \equiv S^*(t)$ splits the domain into early-exercise (stopping) Ω_s and continuation (holding) regions Ω_h $(\Omega = \Omega_s + \Omega_h)$.
- An American option price can be written in terms of European counterpart,

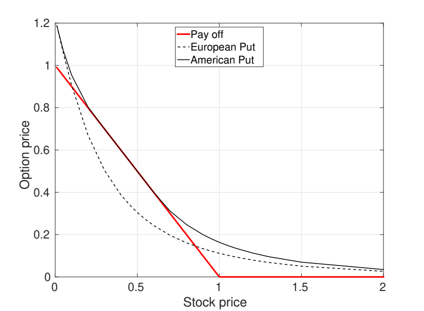
$$V_{am}^P(S_t, t) = \mathbb{E}_t^{\mathbb{Q}}[e^{-r(T-t)} \max(K - S_T, 0)] + \int_t^T \mathbb{E}_u^{\mathbb{Q}}[(rK - qS_u)\mathbf{1}_{\{S_u \in \Omega_s\}}]du.$$

- The second term is the *early-exercise premium*
- At the free boundary, we have,

$$V_{am}(S^*, t) = H(K, S_t^*).$$

Negative rates and dividends

- Two continuation regions may arise when both the interest rate and dividend yield become negative.
- The option value hits the payoff function twice.
- The stopping is between the two early-exercise points.



Settings: $r = -0.01$, $q = -0.06$, $\sigma = 0.2$, $T = 20$, $K = 1.0$.

Implied volatility

- The implied volatility is the level of volatility which, inserted in the pricing model, makes the market and model prices match.
- Implied volatility is an indicator for the future uncertainty of the asset prices as estimated by market participants.
- Computing the implied volatility as an inverse problem:

$$\sigma^* = BS_{am}^{-1}(V_{am}^{mkt}; S, K, \tau, r, q, \alpha),$$

  where $BS_{am}^{-1}(\cdot)$ denotes the inversion of the Black-Scholes formula, and V_{am}^{mkt} is an American option price observed in the market.
- The implied volatility inverse problem is often solved by a nonlinear root-finding method, following an iterative algorithm.
- Given an American option price observed in the market, the implied volatility σ^* is often determined by solving

$$V_{am}^{mkt} - BS_{am}(\sigma^*; S_0, K, \tau, r, q, \alpha) = 0.$$

Issues in computing implied volatility (I)

citic

- Existence of σ^* is guaranteed by the monotonicity of the Black-Scholes equation w.r.t to the volatility in the holding region.
- The option's Vega, becomes zero in the stopping region for American call and put options. It is well-known that,

$$|\Delta| = |\frac{\partial V_{am}}{\partial S}| = 1, \quad \text{Vega} = \frac{\partial V_{am}}{\partial \sigma} = 0.$$

- In other words, the American option prices do not depend on the volatility in the stopping regions. Consequently,

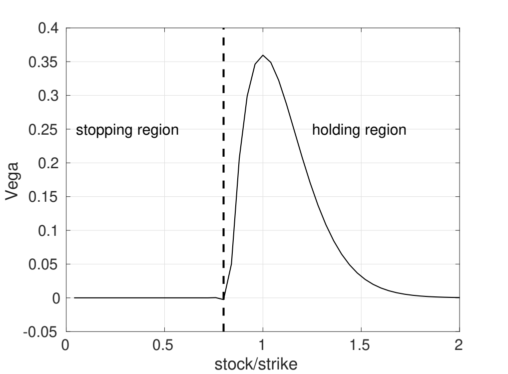
$$\frac{\partial \sigma}{\partial V_{am}} = \frac{1}{\text{Vega}} \to \infty.$$

- When inverting the American Black-Scholes pricing problem in the stopping regions, no unique solution for the implied volatility.

Issues in computing implied volatility (II)

citic

- Therefore, the definition domain should be the continuation region.



The Vega for American puts in different regions.

Implied dividend

citic

- Many companies pay a share of the stock value on the ex-dividend date, which causes the stock price to drop.
- This quantity is often called the actual dividend.
- The *Implied dividend* reflects how the market anticipates future dividend payments of stocks.
- The difference between actual dividends and implied dividends is similar to that between historical and implied volatility. The two parameters reflect different market aspects.
- Some companies do not pay dividends, but the corresponding options may imply a non-zero dividend, which may reflect the borrowing level of the stock.
- The borrowing costs are seen as a factor that influences the implied dividend as a function of the time or the strike price.

Issues computing implied dividend (I)

- Our approach is to estimate implied dividend and implied volatility at once.

- For European options, the implied dividend can be estimated by the put-call parity,

$$V^C_{eu}(S,t) - V^P_{eu}(S,t) = S_t e^{-q\tau} - K e^{-r\tau},$$

  so that,

$$q = -\frac{1}{\tau}\log\left(\frac{V^C_{eu} - V^P_{eu} + K e^{-r\tau}}{S_t}\right).$$

- For American-style options, the put-call parity does not hold.

- What about “de-Americanization” strategy?

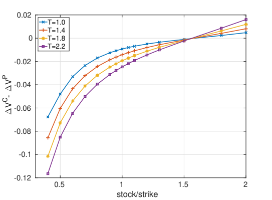
Issues computing implied dividend (II)

- Writting in terms of the early-exercise premiums, ΔV^C and ΔV^P,

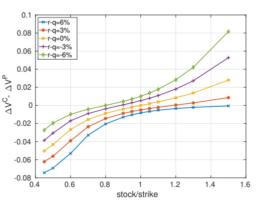
$$V^C_{am} = V^C_{eu} + \Delta V^C \text{ and } V^P_{am} = V^P_{eu} + \Delta V^P.$$

- The "deviation" from the European put-call parity is

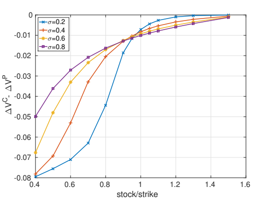
$$\text{EED} := \Delta V^C - \Delta V^P = V^C_{am} - V^P_{am} - S_t e^{-q\tau} + K e^{-r\tau}.$$



(a) EED over maturity T



(b) EED against $r - q$



(c) EED against volatility

Computing the implied dividend

- Again, model-based approach is employed: Black-Scholes.
- The dividend yield is inverted to extract the implied dividend.
- Point of departure is the American call and put model prices:

$$\begin{cases} V_{am}^{C} = BS_{am}(\sigma^{*}, q^{*}; S_0, K, \tau, r, \alpha = 1), \\ V_{am}^{P} = BS_{am}(\sigma^{*}, q^{*}; S_0, K, \tau, r, \alpha = -1), \end{cases}$$

- The system of equations is formulated as a minimization problem.
- Note that a local search based optimization method will most likely not converge when traversing those early-exercise regions.

Artificial Neural Networks (ANN)

citic

- ANNs are powerful function approximators.

- An ANN can be described as a composite function,

$$\mathsf{F}(\mathbf{x}|\boldsymbol{\theta}) = f^{(\ell)}(\ldots f^{(2)}(f^{(1)}(\mathbf{x};\boldsymbol{\theta}^{(1)});\boldsymbol{\theta}^{(2)});\ldots\boldsymbol{\theta}^{(\ell)}),$$

  where $\mathbf{x}$ are the input variables, $\boldsymbol{\theta}$ the hidden parameters (i.e. the weights and the biases in artificial neurons), ℓ the hidden layers, and $f^{(\ell)}(\cdot)$ the activation functions of each layer.

- Once the structure is determined, an ANN becomes a deterministic function.

- And once trained, the the evaluation of *F* is super fast.

Training an ANN

- Determining the values of the hidden parameters which will minimize a loss function.
- A popular approach for training neural networks is to employ first-order optimization algorithms.
- Gradient-based algorithms are often fast, but it may be difficult to calculate the gradients for a large test set.
- Stochastic gradient descent algorithms (SGD) randomly select a portion of the data set (saving memory).
- SGD and its variants (like Adam) are thus preferable to train the ANNs on big data sets.
- In supervised learning, the objective function is

$$\arg\min_{\boldsymbol{\theta}} L(\boldsymbol{\theta}|(\mathbf{X},\mathbf{Y})),$$

given the input-output pairs $(\mathbf{X},\mathbf{Y})$ and a user-defined loss, $L(\boldsymbol{\theta})$.

ANN for implied volatility

- Develop an ANN to approximate the inverse function in Ω_h,

$$\begin{aligned}\sigma^* &= BS_{am}^{-1}(V_{am}^{mkt}; S, K, \tau, r, q, \alpha)\\ &\approx NN(V_{am}^{mkt}; S, K, \tau, r, q, \alpha), \quad [V, S, K, \tau, r, q] \in \Omega_h.\end{aligned}$$

- The ANN must be trained based on the known market variables to approximate the unique target variable σ^*.
- The effective definition domain Ω_h corresponds to the continuation regions.
- The continuation regions are not known initially or are so complicated that there is no analytic formula to describe them.
- However, the counterpart, the early-exercise regions, can be found implicitly from the data.

Computing the holding region

citic

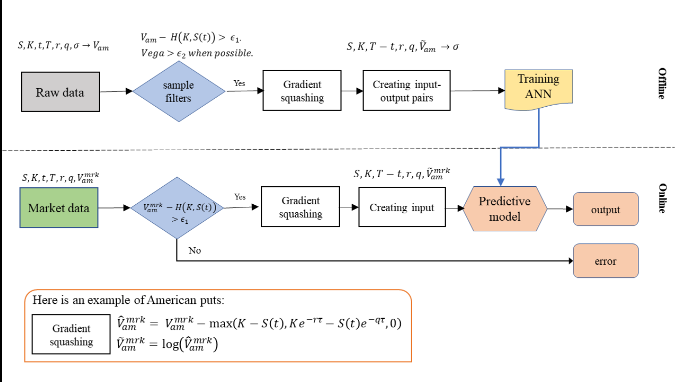
- We obtain the approximate continuation region by $\Omega_h = \Omega - \Omega_s$.
- We use two indicators to detect the samples in Ω_s:
  - The difference between the option value and the payoff.
  - The option's sensitivity Vega.
- For numerical stability, we prescribe some threshold values, ϵ_1 and ϵ_2:

$$|V_{am}(S_t, K, \tau, r, q, \sigma) - H(K, S_t)| > \epsilon_1.$$

$$\text{Vega} > \epsilon_2.$$

- The first one should cover the second one, but for robustness reason, both are enforced.
- The procedure is done off-line, prior the training phase, by filtering out the samples according to the criteria.

ANN for implied volatility



A flowchart of the ANN-based method to compute the implied volatility from American option prices.

ANN for implied information

- Determining both implied volatility and implied dividend simultaneously.
- We assume the implied volatility and the implied dividend are identical for calls and puts (given K, S_0, T, t and r):

$$\begin{cases} V_{am}^{C,mkt} - BS_{am}(\sigma^*, q^*; S_0, K, \tau, r, \alpha = 1) = 0, \\ V_{am}^{P,mkt} - BS_{am}(\sigma^*, q^*; S_0, K, \tau, r, \alpha = -1) = 0, \end{cases}$$

- Then, we have two unknown parameters to calibrate, implied volatility σ^* and implied dividend yield q^*, from a pair of American option prices, $V_{am}^{C,mkt}$ and $V_{am}^{P,mkt}$.
- The above system is reformulated as a minimization problem,

$$\arg \min_{\sigma^* \in R^+, q^* \in R} (BS_{am}(\sigma^*, q^*; \alpha = 1) - V_{am}^{C,mkt})^2 + (BS_{am}(\sigma^*, q^*; \alpha = -1) - V_{am}^{P,mkt})^2.$$

- We adapt a fast, generic and robust calibration framework, the CaNN (Calibration Neural Networks) developed in [1].

CaNN for implied information

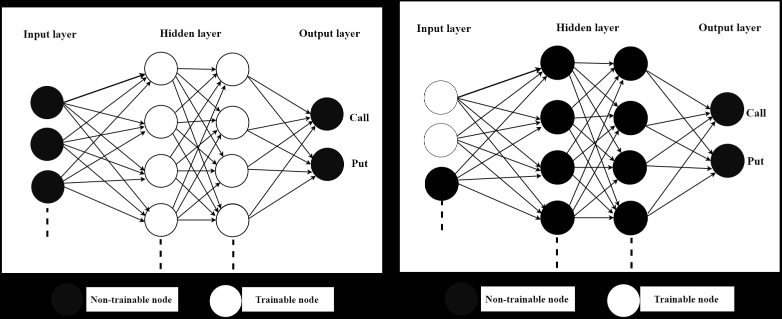
- The CaNN consists of two stages:
  - The forward pass, including the training and testing phase, approximates the American-style Black-Scholes prices.
  - The backward pass finds the two parameters (σ^*, q^*) to match the two observed American option prices, $V_{am}^{P,mkt}$ and $V_{am}^{C,mkt}$ (given K, T, S_0, r).

- We have developed one neural network providing two output values, the American call and put prices.
- The objective function is written as

$$\arg \min_{\sigma^* \in R^+, q^* \in R} (\mathrm{NN}(\sigma^*, q^*; \alpha = 1) - V_{am}^{C,mkt})^2 + (\mathrm{NN}(\sigma^*, q^*; \alpha = -1) - V_{am}^{P,mkt})^2,$$

which is used as the loss function for the backward pass.

CaNN structure



(a) Training phase (b) Calibration phase

Left: In the forward pass of the CaNN, the output layer produces two option prices. Right: In the calibration phase, the CaNN estimates the two parameters, implied volatility and implied dividend, in the original input layer.

The ANN design and training set

citic

- We find a balance between representation power and efficiency with the following configuration:

Hyper-parameter	Value
Hidden layers	4
Neurons (each layer)	200
Activation	Softplus
Initialization	Glorot_uniform
Optimizer	Adam
Batch size	1024

The ANN configuration.

- Other useful operations for deep NN (dropout or batch normalization) do not bring any significant benefits in our “shallow” ANN.
- Samples for the training/test set are generated by the COS method.

Settings for computing implied volatility

- Without loss of generality, we use a fixed spot price $S_0 = 1.0$.
- The two thresholds are set to $\epsilon_1 = 0.0001$ and $\epsilon_2 = 0.001$.
- We consider the measures

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2, \quad \text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|, \quad \text{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\frac{|y_i - \hat{y}_i|}{y_i}.$$

ANN	Parameters	Value range	Employed method
ANN Input	Strike, K	[0.6, 1.4]	LHS
	Time value, $\log(\hat{V}^P_{am})$	$(-11.51, -0.24)$	COS
	Time to maturity, τ	[0.05, 3.0]	LHS
	Interest rate, r	[-0.05, 0.1]	LHS
	Dividend yield, q	[-0.05, 0.1]	LHS
ANN output	Implied volatility, σ^*	$(0.01, 1.05)$	LHS

Train dataset for American options under the Black-Scholes model; The spot price $S_0 = 1$ is fixed. The upper bound of American put price is 1.2. LHS stands for Latin Hypercube Sampling.

Numerical results: implied volatility

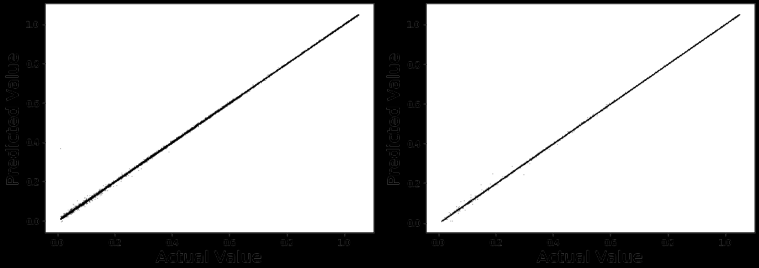
- We analyse the training and testing performance.
- The test performance is close to the train performance, suggesting that the trained ANN generalizes well for unseen data
- The ANN predicted implied volatility values approximate the true values accurately for both the train and test datasets, as is indicated by the R^2 measure.

-	MSE	MAE	MAPE	R^2
Training	$4.33 \cdot 10^{-7}$	$2.44 \cdot 10^{-4}$	$1.11 \cdot 10^{-3}$	0.999994
Testing	$4.60 \cdot 10^{-7}$	$2.51 \cdot 10^{-4}$	$1.15 \cdot 10^{-3}$	0.999993

Multiple measures are used to evaluate the performance.

Numerical results: implied volatility

- It is observed that the trained model performance tends to decrease when the pricing model parameters gets close to the upper or lower bounds
- Thus the training data set is recommended to have a wider parameter range than the test range of interest.



(a) Training (b) Testing

Left: R^2=0.999994; Right: R^2=0.999993

Settings for computing implied information

citic

- The loss function includes two components:

$$\text{MSE} = \frac{1}{2n}\sum_{i=1}^{n}\{(\tilde{V}^{P}_{am,i} - V^{P,mod}_{am,i})^2 + (\tilde{V}^{C}_{am,i} - V^{C,mod}_{am,i})^2\}.$$

- Training data set for the forward pass:

ANN	Parameters	Value range	Method
Forward input	Strike, K	[0.45, 1.55]	LHS
	Time to maturity, τ	[0.08, 3.05]	LHS
	Risk-free rate, r	[-0.1, 0.25]	LHS
	Dividend yield, q	[-0.1, 0.25]	LHS
	Implied volatility, σ	$(0.01, 1.05)$	LHS
Forward output	American put, V^{P}_{am}	$(0, 1.8)$	COS
	American call, V^{C}_{am}	$(0, 1.2)$	COS

We fix $S_0 = 1$, and sample strike prices K to generate different moneyness levels. The total number of the data samples is nearly one million, with 80% training, 10% validation, 10% test samples.

CaNN: performance of the forward pass

- The results, for both Calls and Puts, are highly satisfactory, achieving very good levels of precision in all the considered measures.

–	Option	MSE	MAE	MAPE	R^2
Training	Call	1.40×10^{-7}	3.00×10^{-4}	1.25×10^{-3}	0.9999965
	Put	2.54×10^{-7}	4.24×10^{-4}	1.64×10^{-3}	0.9999959
Testing	Call	1.43×10^{-7}	3.02×10^{-4}	1.27×10^{-3}	0.9999964
	Put	2.55×10^{-7}	4.26×10^{-4}	1.64×10^{-3}	0.9999959

The performance of the CaNN forward pass with two outputs.

CaNN: performance of the backward pass

citic

- The results suggest that the CaNN can accurately recover the implied volatility and implied dividend from “artificial market option data”.
- Even in complex scenarios (negative interest rates and/or dividend yields), CaNN recovers the true values.

K/S_0	T	r	$\sigma^\dagger$	$q^\dagger$	C_{am}^{mkt}	P_{am}^{mkt}	σ^*	q^*
1.0	0.5	-0.04	0.1	0.06	0.0146	0.0597	0.099	0.059
1.1	0.5	-0.04	0.2	-0.06	0.0255	0.1181	0.198	-0.061
1.0	0.75	0.0	0.3	-0.02	0.1119	0.0976	0.300	-0.020
1.2	1.0	-0.04	0.4	0.08	0.0603	0.3810	0.40	0.080
0.8	1.0	0.02	0.3	0.02	0.2322	0.03472	0.299	0.020
0.7	1.25	0.0	0.4	-0.04	0.3886	0.0378	0.399	-0.040

Using CaNN to extract implied volatility and implied dividend. $\dagger$ indicates the prescribed values, $*$ indicates the calibrated values.

Conclusions

- We studied a data-driven method to extract the implied volatility and/or implied dividend yield from observed market American option prices in a fast and robust way.
- For computing the American implied volatility, we propose a sophisticated ANN to approximate the inverse problem.
- The problem domain is extracted from the data, preserving the ANN offline-online decoupling advantage.
- We also propose a method for finding simultaneously implied dividend and implied volatility from American options using a calibration approach.
- The numerical experiments demonstrate that the CaNN is able to accurately extract multiple pieces of implied information from American options.
- It should be feasible to extend the approach to deal with time-dependent or discrete dividends.

References

[1] Shuaiqiang Liu et al. “A neural network-based framework for financial model calibration”. In: *Journal of Mathematics in Industry* 9.1 (2019), p. 9. ISSN: 2190-5983.

[2] Shuaiqiang Liu et al. “On a Neural Network to extract implied information from American options”. In: *Applied Mathematical Finance* 28.5 (2021), pp. 449–475.

[3] Bowen Zhang and Cornelis W. Oosterlee. “Efficient pricing of European-style Asian options under exponential Lévy processes based on Fourier cosine expansions”. In: *SIAM Journal on Financial Mathematics* 4.1 (2013), pp. 399–426.

Acknowledgements & Questions

Thanks to the support received from the Centro de Investigación de Galicia “CITIC”, funded by Xunta de Galicia and the European Union (European Regional Development Fund- Galicia 2014-2020 Program), by grant ED431G 2019/01.

More: `alvaro.leitao@udc.gal` and `alvaroleitao.github.io`

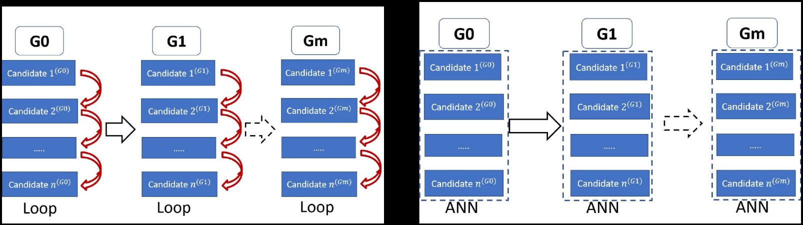
Gracias!

CaNN for implied information

citic

- We adapt a fast, generic and robust calibration framework, the CaNN (Calibration Neural Networks) developed in [1].
- The basic idea of the methodology is to convert the calibration of model parameters into an estimation of a neural network's hidden units.
- The model calibration and training ANNs can be reduced to solving an optimization problem.
- It enables parallel GPU computing to speed up the computations, which makes feasible to employ a global optimization technique to search the solution space.
- Here, we employ the gradient-free optimization algorithm, Differential Evolution (DE) which does not get stuck in local minima or in the stopping region.
- And DE is an inherently parallel technique.

CaNN for implied information



(a) Conventional DE (b) Parallel DE

The global optimizer DE runs in parallel within CaNN. "Gm" represents the *m*-th generation, where there are *n* candidates of to-be-optimized parameters, i.e., *n* sets of open parameters in the pricing model. These *n* sets of model parameters are independent, thus can be processed by the ANN simultaneously, instead of a for-loop, to reduce the computation time.