Computations in Option Pricing Engines

Vital Mendonca Filho, Pavee Phongsopa, Nicholas Wotton

Advisors: Yanhua Li, Qingshuo Song, Gu Wang

April 24, 2020

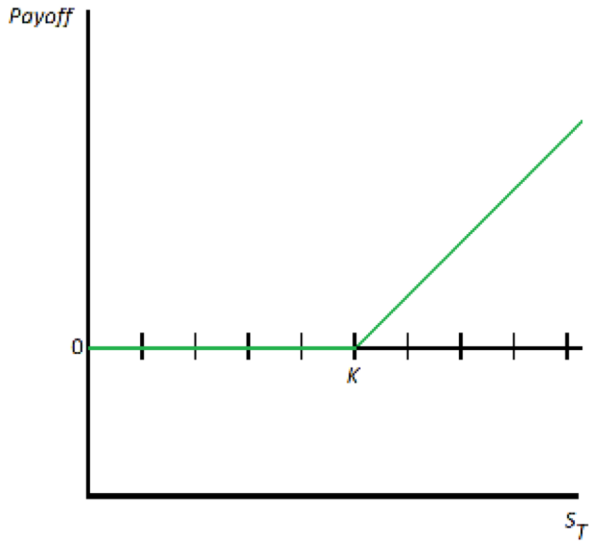
Outline

- Traditional Methods in Option Pricing
- Exploring Modern Approach
- Computations of Option Prices via Neural Network
- Applications: Greeks
- Conclusion

Brief Overview of Options

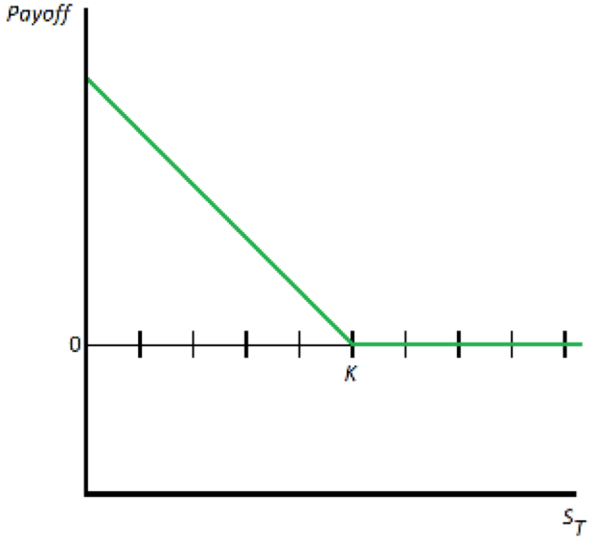
- Call

$$C(T) = \max[0,\ S_T - K]$$



- Put

$$P(T) = \max[0,\ K - S_T]$$

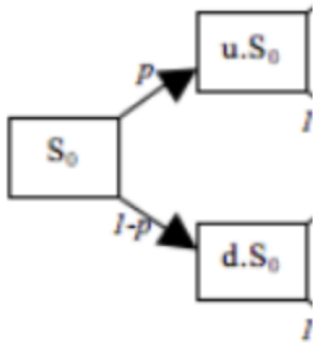


- European vs American

CRR Model

- Recursive model prices of options contracts in discrete time
- Option price at end of each node (Call or Put)
- Uses Binomial Tree

$$C_{t-\Delta t,i} = e^{-r\Delta t}(pC_{t,i} + (1-p)C_{t,i+1})$$



$$u = e^{\sigma\sqrt{t/n}}$$

$$d = e^{-\sigma\sqrt{t/n}}$$

$$p = \frac{e^{rt/n} - d}{u - d}$$

Wikipedia.Binomial Options Pricing Model

Black-Scholes Model (BSM)

- Mathematical model to price options contracts in continuous time
- The option prices with maturity T and strike price K
  - The Model

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS\frac{\partial V}{\partial S} - rV = 0$$

$$C_0 = S_0\Phi(d_1) - Ke^{-rT}\Phi(d_2),$$

and

$$P_0 = Ke^{-rT}\Phi(-d_2) - S_0\Phi(-d_1),$$

where

$$d_1 = \frac{1}{\sigma\sqrt{(T-t)}}\left[\ln\frac{S_0}{K} + \left(r + \frac{\sigma^2}{2}\right)(T-t)\right], \quad d_2 = d_1 - \sigma\sqrt{T-t},$$

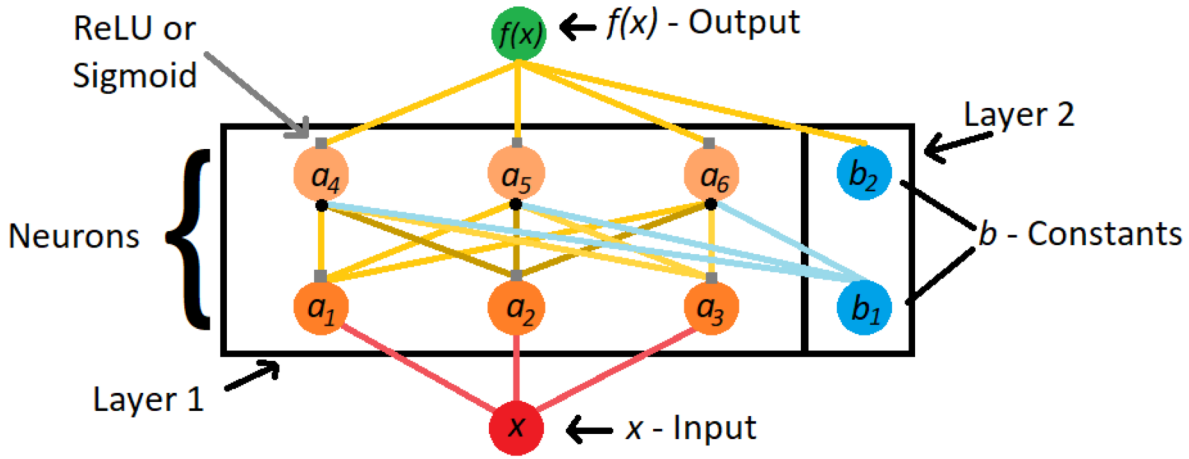
$$\Phi(x) = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{x} e^{-\frac{t^2}{2}}\,dt.$$

Steven Shreve. Stochastic Calculus for Finance II: Continuous Time Models. Springer Finance, 2010.

Neural Networks

- Approximates Complex Function
- Neurons: Simple Functions
- Activation Functions

$$N(x) = \phi_2 \left(\left(\phi_1 \left([x] \begin{bmatrix} a_{11} & a_{12} & a_{13} \end{bmatrix} \right) + \begin{bmatrix} b_1 & b_1 & b_1 \end{bmatrix} \right) \begin{bmatrix} a_{21} \\ a_{22} \\ a_{23} \end{bmatrix} \right) + b_2$$

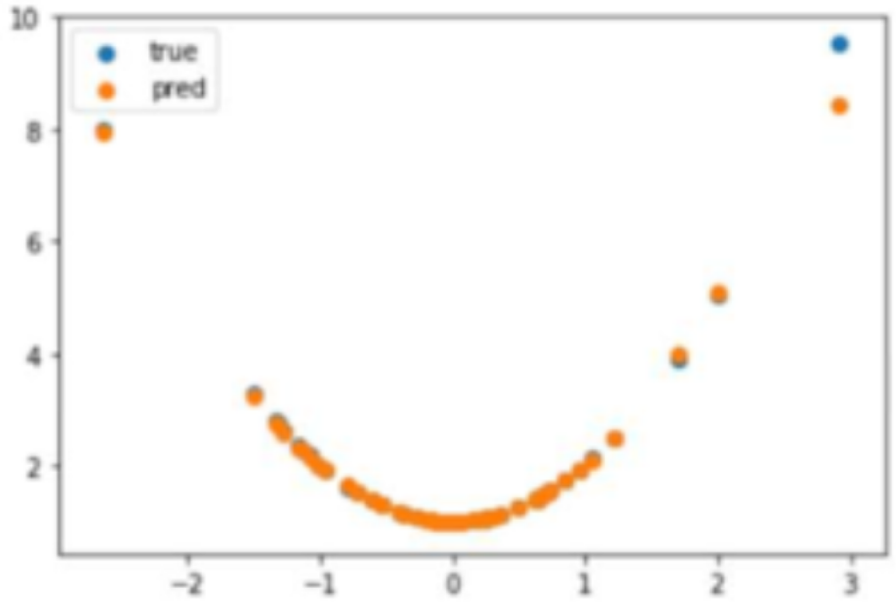


A Simple Neural Network

Activation Functions

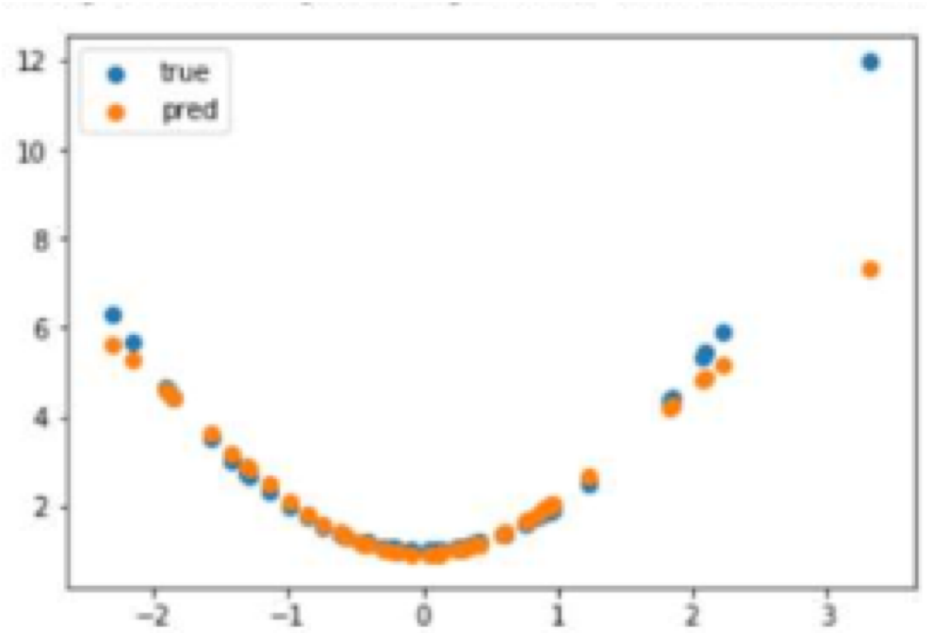
- ReLU

$$ReLU : \phi(x) = \max\{0, x\}$$



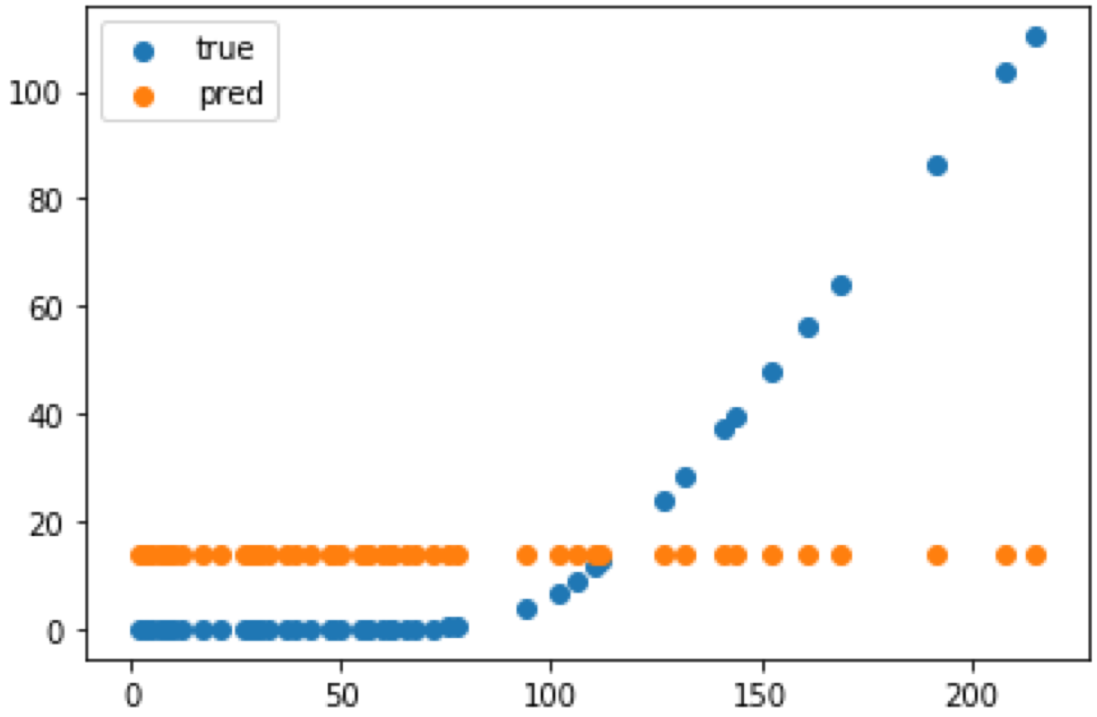
- Sigmoid

$$Sigmoid : \phi(x) = \frac{1}{1 + e^{-x}}$$



BSM With Neural Network

- Underlying Price vs Option Price
- Model Malfunction

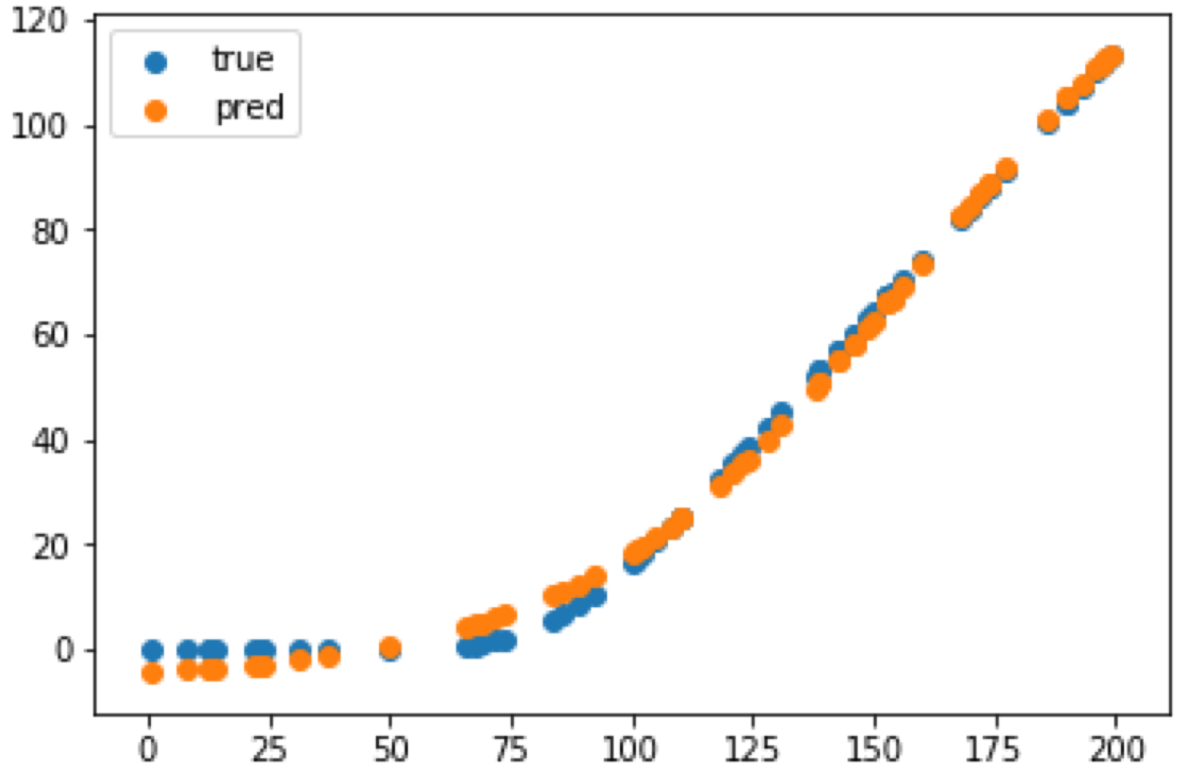


True vs Predicted Values for a European Call Option

Normalization

- Attempt to Fix
- *u, l* - scale parameters
  - *u* = 1, *l* = 0
  - *u* = 1, *l* = -1

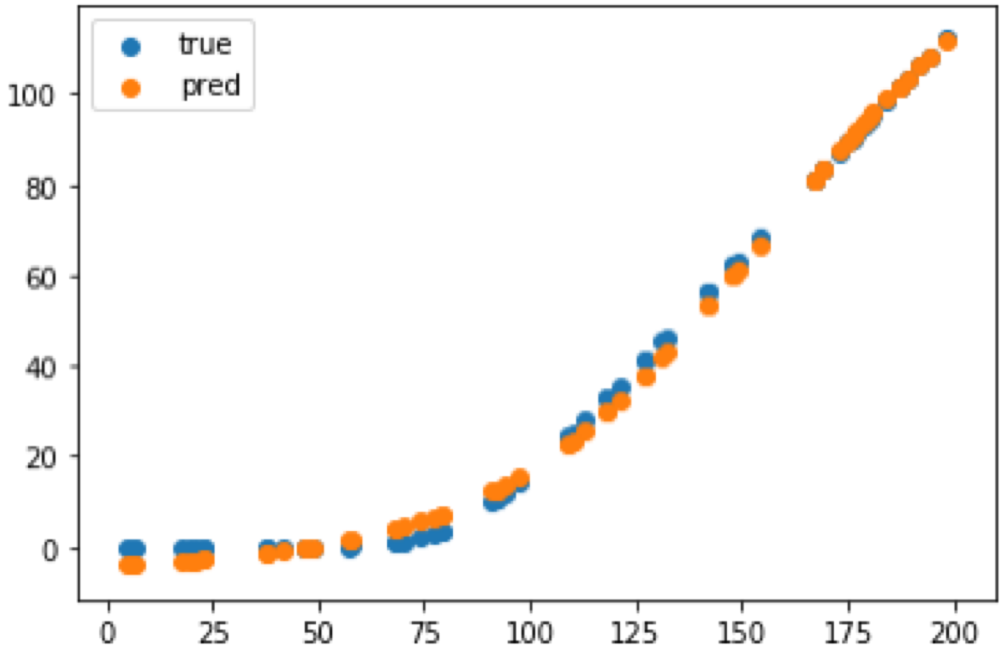
$$x_N = \frac{u - l}{\max(x) - \min(x)}(x - \min x) + l,$$



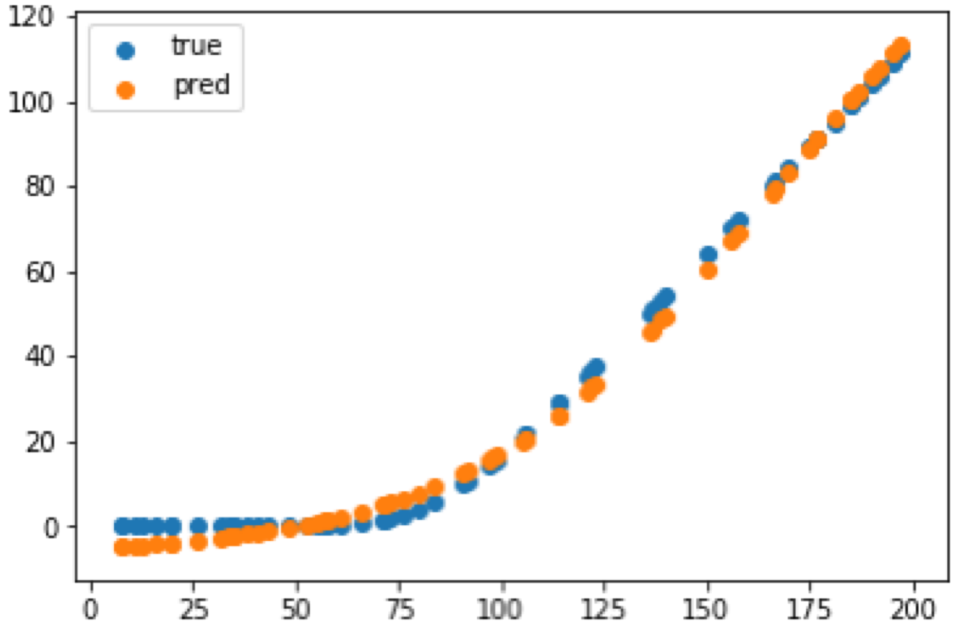
True vs Predicted Values for a European Call Option
50 Neurons Layer 1 and 12 Neurons Layer 2

BSM Neural Network - Further Exploration

- Varied Neurons
- Varied Layers



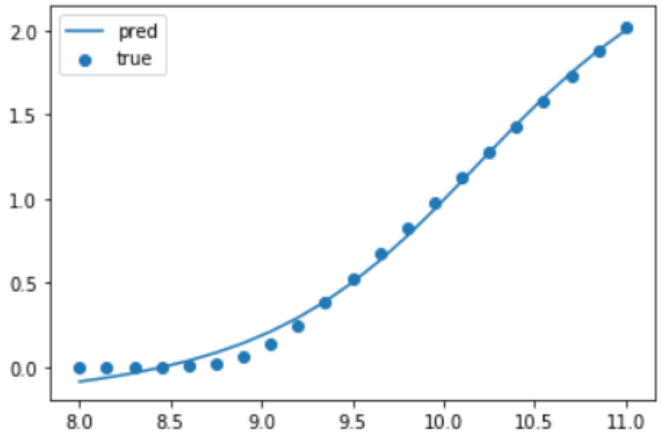
100 Neurons Layer 1; 100 Neurons Layer 2
Loss: 0.00052513
Time: 4.812922478 Seconds

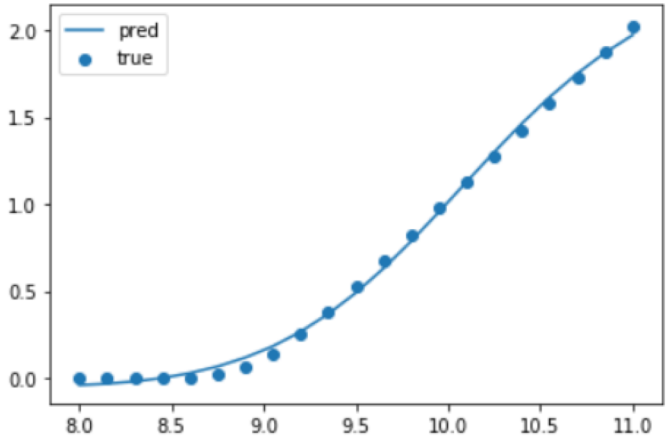


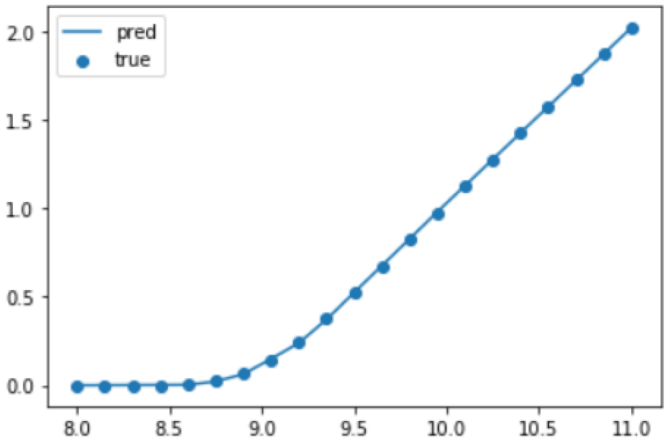
50 Neurons Layer 1; 50 Neurons Layer 2; 100 Neurons Layer 3
Loss: 0.00111636
Time: 5.457895279 Seconds

Universal Approximation Theorem

Theorem A feed-forward network with a single hidden layer containing a finite number of neurons can approximate continuous functions on compact subsets of Rn, under mild assumptions on the activation function



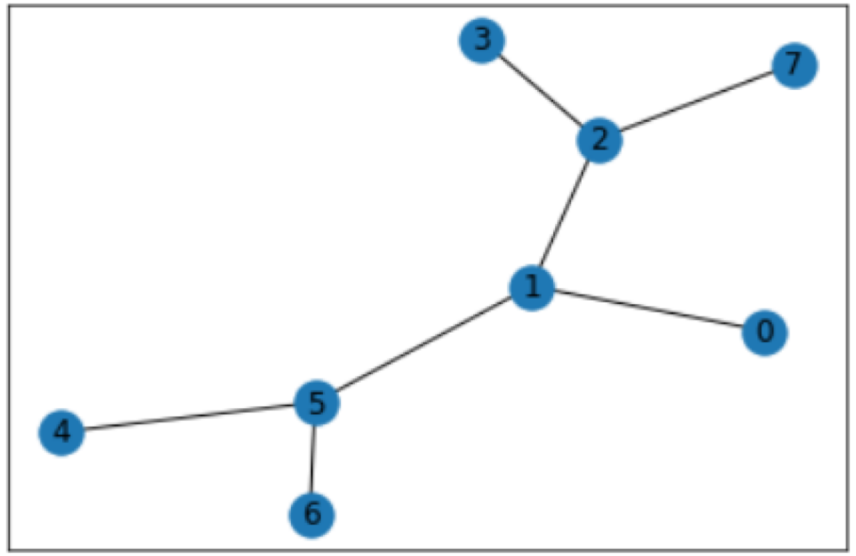




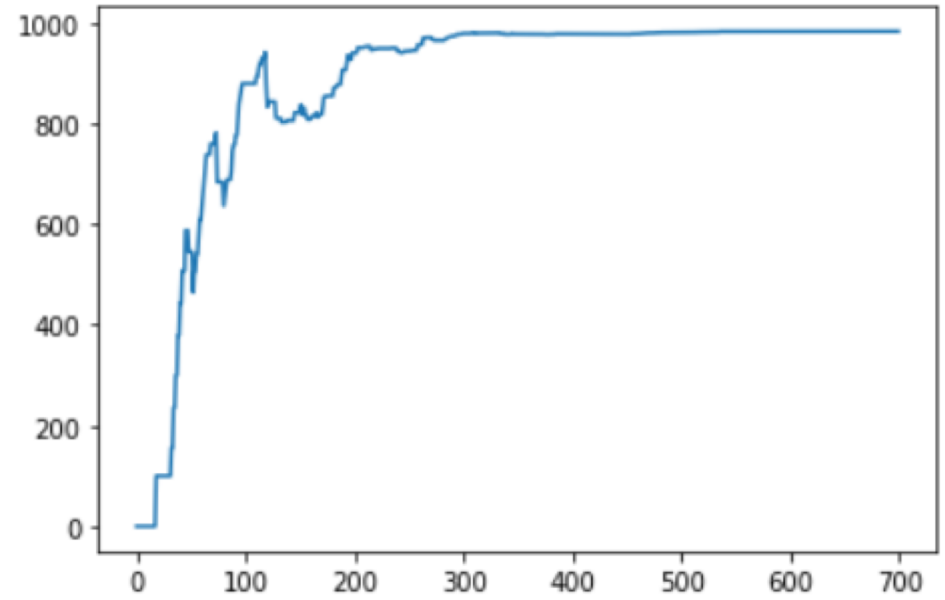
Wikipedia.Universal Approximation Theorem

Q-Learning

- Model Free Reinforcement Learning
- Reward Based Algorithm




```
Most efficient path:
[0, 1, 2, 7]
```



Greeks

- Collection of partial derivatives of the BSM model with respect to:
  - Delta → price of stock
  - Theta → time to expiration
  - Rho → risk-free interest rate
  - Vega → volatility

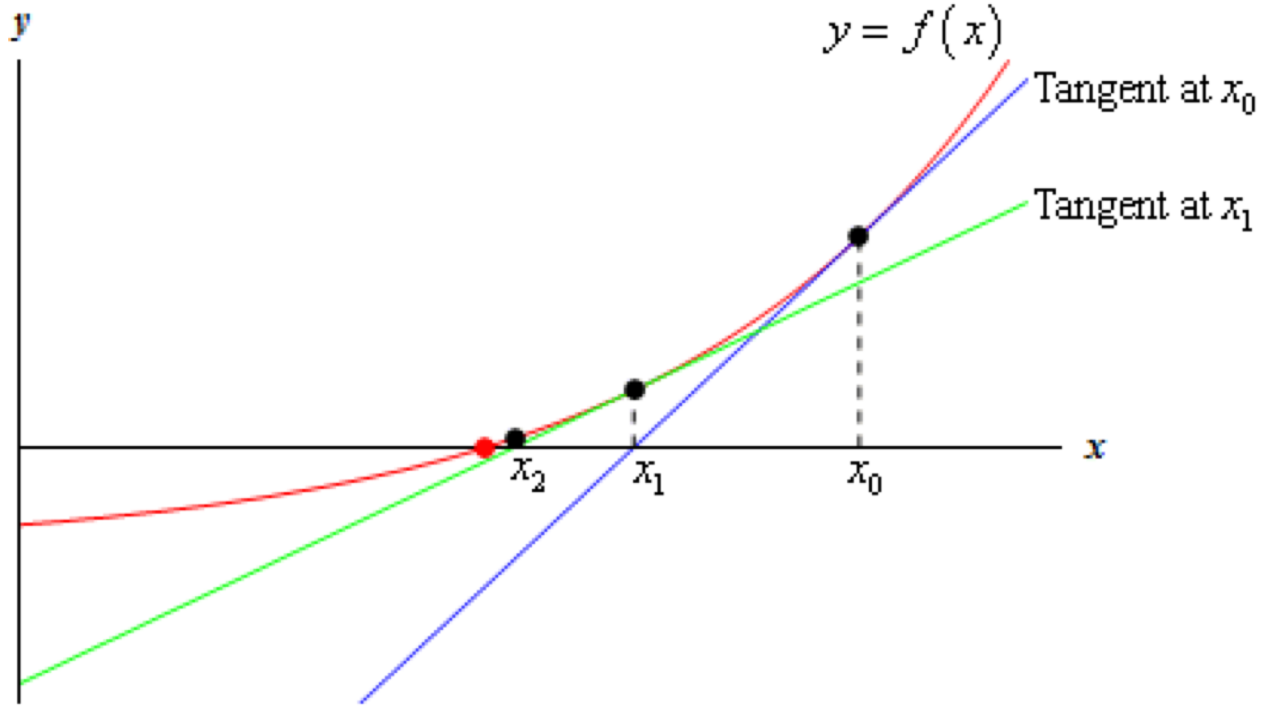
- Vega does not have a closed form solution
  - Future work can exploit approximation of vega using machine learning

Implied Volatility

- Market's expected volatility for an asset during the contract's duration
  - Extracted from the market price of the contract
- Volatility Smile
  - Deeper out-the-money and in-the-money contracts have higher implied vol
- Traditionally approximated by Newton's Method

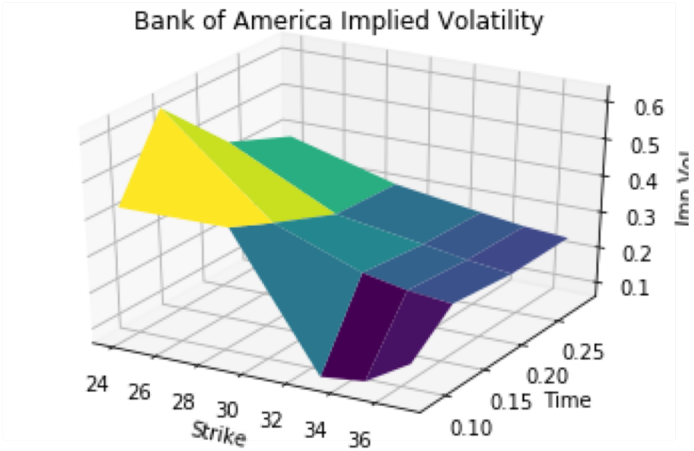
Newton's Method

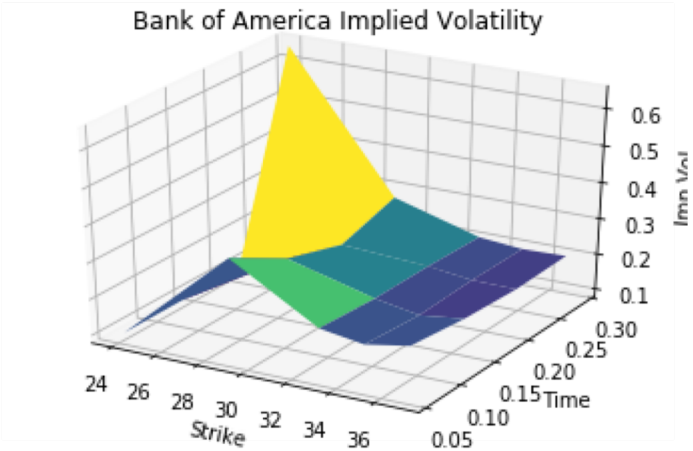
- Recursive method to approximate the root of a differentiable function
- We approximate the implied volatility using BSM and Vega



Volatility Surface

- Surface generated from comparing Implied Volatility with different times to maturity and strikes





Conclusion

- Explored Machine Learning Techniques
- Implemented Neural Networks in Financial Models