Financial Pricing Engine

Vital Mendonca Filho(MA), Pavee Phongsopa(MA), Nicholas Wotton(MA)

1st-Advisor: Song Qinghuo(Department of Mathematical Science)

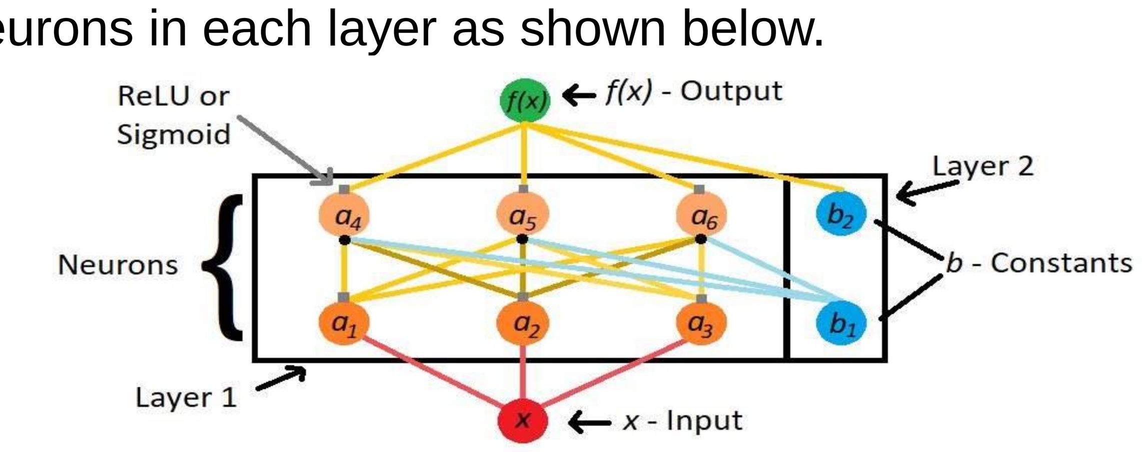
2nd-Advisor: Gu Wang (Department of Mathematical Science)

Abstract

The advancement of computing power has allowed mathematicians, statisticians, computer scientists and other professionals to solve complex problems using powerful machines. The progress in data science, more specifically in deep learning opened ways to solve high dimensional equations, bypassing the curse of dimensionality, a phenomena which illustrates the complexity of equations as they grow in degree. In this project we seek to use deep learning tools to price exotic financial products, combining stochastic partial differential equation and neural networks.

Background

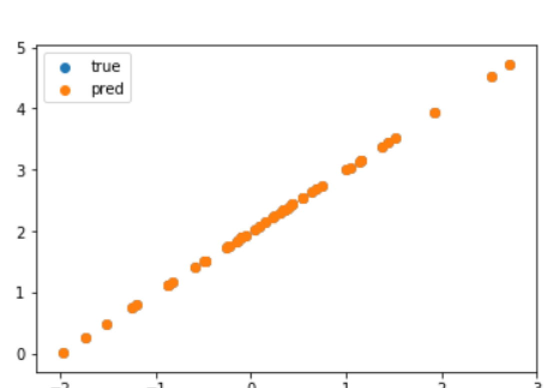
Deep learning refers to the study of large, complex data sets through neural networks. The idea is that we feed the system a functions and neurons in deep layers will be able to interpret a result for such input after extensive ”training”. A typical model usually consists of a few neurons in each layer as shown below.



$$\left(\begin{bmatrix} x \end{bmatrix}\begin{bmatrix} a_1 & a_2 & a_3 \end{bmatrix} + \begin{bmatrix} b_1 & b_2 & b_3 \end{bmatrix}\right)\begin{bmatrix} a_4 \\ a_5 \\ a_6 \end{bmatrix} + \begin{bmatrix} b_4 \end{bmatrix} = \begin{bmatrix} f(x) \end{bmatrix}$$

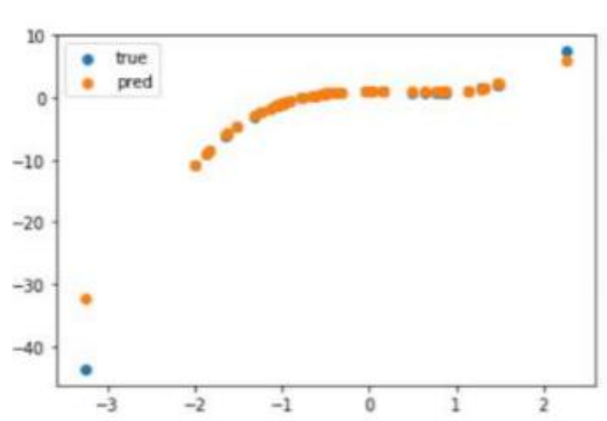
Approximation

Data show promising result in approximating linear function and varying degree polynomials.




```
model = nn.Sequential(
    nn.Linear(1, 3),
    nn.ReLu(),
    nn.Linear(3, 1)
)
```

```
model = nn.Sequential(
    nn.Linear(1, 20),
    nn.ReLu(),
    nn.Linear(20, 20),
    nn.Linear(20, 20),
    nn.Linear(20, 1)
)
```



BS Model

BSM or Black-Scholes Model is a mathematical model use for the derivation of option value with partial differential equation in the model known as Black-Scholes equation. The formula returns a theoretical estimate of the European style option.

Black-Scholes model assume the distribution of stock as lognormal that writes:

$$\ln\frac{S(T)}{S(0)} \sim \mathcal{N}\left((r - \tfrac{1}{2}\sigma^2)T,\ \sigma^2 T\right)$$

Call Option:

$$C_0 = \mathbb{E}[e^{-rT}(S(T)-K)^+] = S_0\Phi(d_1) - Ke^{-rT}\Phi(d_2),$$

Put Option:

$$P_0 = \mathbb{E}[e^{-rT}(S(T)-K)^-] = Ke^{-rT}\Phi(-d_2) - S_0\Phi(-d_1),$$

With d_1 and d_2:

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

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

Implied Volatility

Option Pricing such as Black-Scholes model or CRR model requires variety of inputs to calculate the theoretical value of an option which includes the value of volatility. In most cases, volatility is not known. Therefore, we must also know how to calculate the expected volatility value from historical data with the following formula.

$$vega(C) = \frac{1}{\sqrt{2\pi}} S e^{-q(T-t)} e^{\frac{-d_1^2}{2}} \sqrt{T-t}$$

CRR Model

Also know as Binomial Option Pricing model. Like Black-Scholes Model, the model produces a theoretical option value given a set of data. However, the model is broken down into number of “steps” which is further broken down into a binomial tree of “up” and “down.” At each step, the model predicts two possible outcomes each with certain possibility of happening base on the time and volatility.

The model consists of 3 major steps

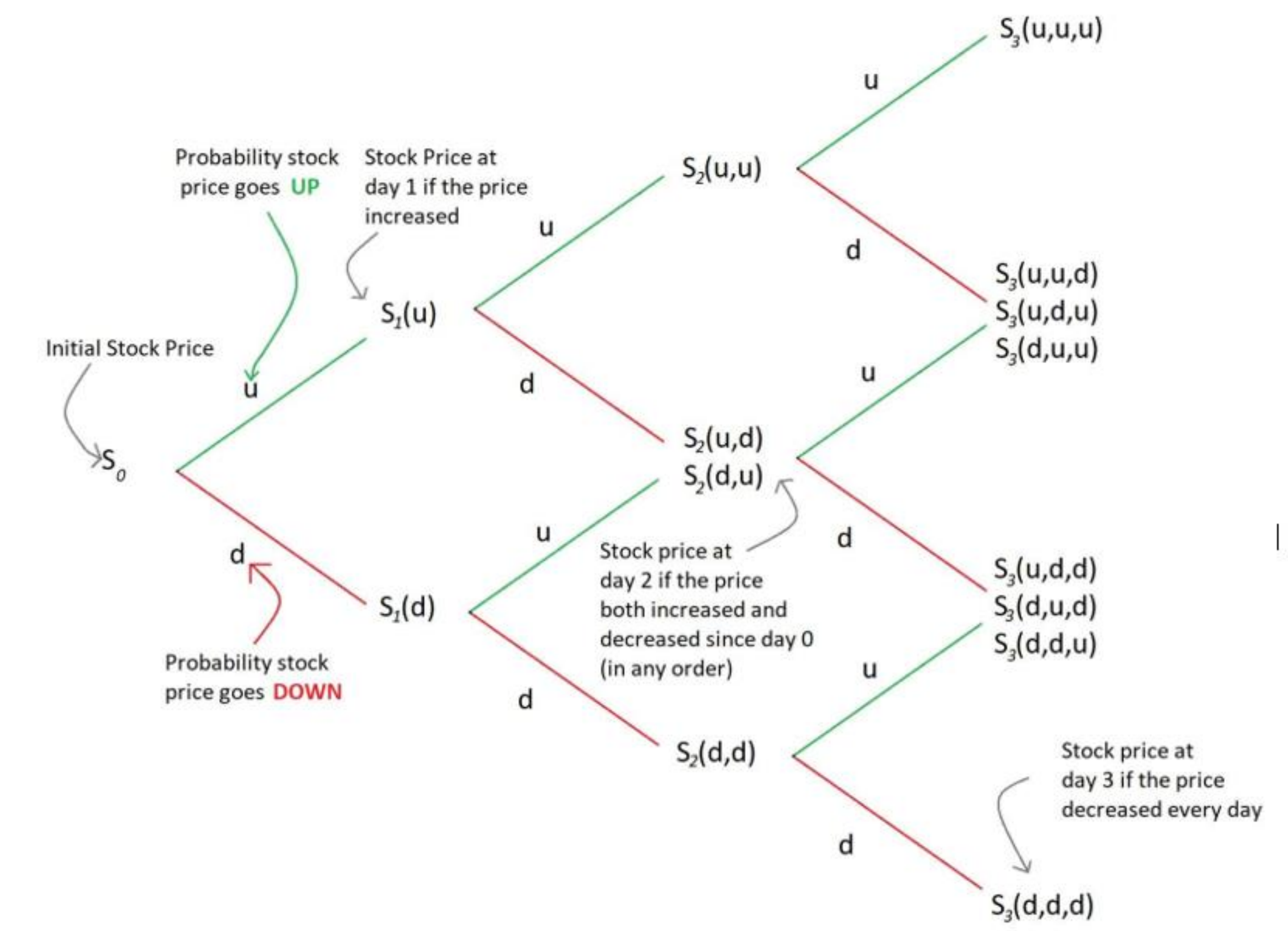
1. Creating Price Tree with price(p), up(u), and down(d)

$$p = \frac{e^{rt/n} - d}{u - d} \qquad u = e^{\sigma\sqrt{t/n}} \qquad d = e^{-\sigma\sqrt{t/n}}$$

2. Find Option Value at each final nodes

Max [$(S_n - K)$, 0], for a call option
Max [$(K - S_n)$, 0], for a put option:

3. Calculating backward to find option value of all the nodes according to the type of option (Europeans, American, Bermudan, etc.)



References

- https://en.wikipedia.org/wiki/Binomial_options_pricing_model
- https://aaronschlegel.me/implied-volatility-functions-python.html
- https://en.wikipedia.org/wiki/Black%E2%80%93Scholes_model
- http://www.deeplearningbook.org/
- https://github.com/songqsh/MQP2019/blob/master/doc/HJE18_deep_bsde.pdf
- https://github.com/songqsh/MQP2019/blob/master/doc/EHJ17_deep_bsde.pdf