Machine Learning
Project Proposal

Brady Metherall

Tuesday October 31, 2017

The goal of this project is to forecast stock prices using a deep neural network. While a daunting task, our first goal is to simulate a toy stock market in which we have total control of the parameters, and trends. This will better allow us to set up, train, test, and debug the neural network.

To do this we shall use the Black-Scholes model. In 1973 Fischer Black and Myron Scholes modelled stock prices with geometric Brownian motion, and Robert Merton extended its applicability. In 1997 they were awarded the Nobel Prize in Economics for their work. The price of a stock, S_t, is modelled by the stochastic differential equation

$$dS_t = \mu S_t dt + \sigma S_t dW_t, \tag{1}$$

Figure 1: Sample stock data generated from (2).

where W_t is a Wiener process. This can be discretized as

$$S_{t+1} = S_t \left(1 + \mu \Delta t + \sigma \mathcal{N}(0,1)\sqrt{\Delta t} \right), \tag{2}$$

and this algorithm can easily be implemented, and an example of the result can be seen in Figure 1. From Itô calculus the analytic solution to (1) is

$$S_t = S_0 \exp \left(\left(\mu - \frac{1}{2}\sigma^2 \right) t + \sigma W_t \right)$$

which may be useful for finding trends.

In probability theory, the characteristic function is an alternative representation of a probability distribution. The characteristic function, $\phi(\lambda)$, is given by $\mathbb{E}[\exp(i\lambda X_t)]$, and as such is equal to the Fourier Transform of X_t. Working in the Fourier space provides a frequency representation of the stock prices, which may help remove the short-term noise in the prices and extract the long-term trends.

Generally, the characteristic function of a probability distribution is written in the form of the Lévy-Khintchine representation

$$\phi(\lambda) = \exp \left(i\mu\lambda - \frac{\lambda^2 \sigma^2}{2} + \int_{\mathbb{R}} e^{i\lambda x} - 1 - i\lambda x 1_{\{|x|<1\}} \nu(dx) \right).$$

The first two terms come from Brownian motion with drift whereas the latter term comes from a Poisson jump. In the case $\nu = 0$, the last term vanishes and we simply obtain the characteristic function of the
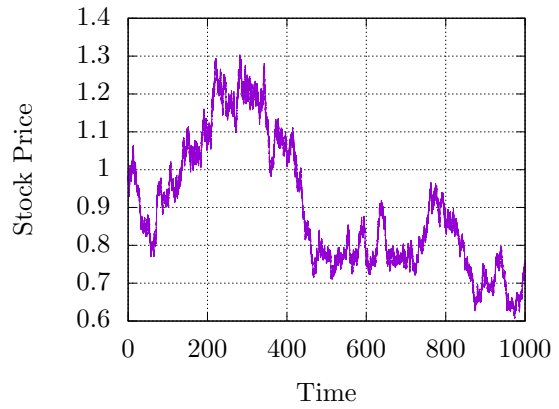
Black-Scholes model. With the characteristic function we have two different representations of the stock prices which we can use as a form of cross-validation for regression and classification.

Lastly, our final goal is to apply our neural network to real world stocks. The `pandas` package in Python includes functions to automatically pull financial data from Yahoo Finance which can then be divided into, for example, bi-weekly intervals and used to train and test the network. The Yahoo Finance data includes the daily high, low, open, and close prices, as well as the volume traded for each business day for the last five years. By looking at stocks within a particular industry, we should be able to obtain more than enough data in order to train the neural network.