Project Summary – Neural Network Approaches to PDEs in Option Pricing

Annie Zhang

I began by reproducing the results from the original BSDE paper and extended the method to test whether we could recover the full option price surface across time. In the SABR setting, I successfully implemented the BSDE framework to learn the delta process, adapting it to this more complex stochastic volatility model. However, I soon realized that the BSDE formulation is fundamentally limited to outputting the option price at $t = 0$ and cannot recover $u(t, S)$ across arbitrary time steps. Additionally, any change in parameters like r or σ required retraining the entire model, which made it less practical for generalized pricing scenarios.

Given these limitations, I suggested we explore the Deep Galerkin Method (DGM), as its direct PDE-based formulation appeared more flexible and generalizable. We initially reused the same network architecture that had been used for BSDE, but quickly encountered vanishing gradient issues that hindered training. After analyzing the problem, I proposed and designed a deeper ResNet-style architecture to improve gradient flow and model stability. This change significantly enhanced training performance. With the new architecture, we were able to recover the full option price surface $u(t, S; r, \sigma, K)$ in the Black-Scholes case and $u(t, F; \alpha, r, \nu, \rho, \beta)$ in the SABR setting. It also allowed the model to generalize to new parameter values without retraining, making it more efficient for practical applications.

We then attempted to apply DGM to solve the Fokker-Planck equation for the Black-Scholes model. However, the delta-function initial condition made it difficult for the model to capture the sharp peak in the density at early time steps, and the training proved unstable. To address this, I reviewed additional literature and proposed adapting a reference-based neural network approach, where low-accuracy Monte Carlo samples are used as reference points to guide the learning of the density function. This idea significantly improved both training stability and convergence speed. Building on this, we modeled each time-step distribution individually using a Mixture Density Network (MDN), which allowed the model to accurately capture the sharp log-normal profiles observed at short maturities. As a result, we adopted the Sliced MDN framework for solving the Fokker-Planck equation under the Black-Scholes setting.