# Project #2: Black-Scholes Case Study

**Objective:** This project aims to develop a neural network model to approximate the nonlinear function of the Black-Scholes (BS) formula for financial option derivatives. The study involves experimenting with various neural network parameters and evaluating model performance based on mean squared error (MSE).

**Part A: Neural Network Architecture and Hyperparameter Tuning**

**Objective:** Evaluate the impact of varying network layers, activation functions, number of neurons, optimizers, and training epochs to determine the optimal configuration for the Black-Scholes approximation.

**1. Varying Layers**

We experimented with three architectures by modifying the number of layers: a shallow network with 1 layer and deeper networks with 2 and 3 layers. The evaluation metric for each architecture was Mean Absolute Error (MAE) on the test set.

* **Results:**
  + **1 layer:** MAE = 0.1851
  + **2 layers:** MAE = 0.1926
  + **3 layers:** MAE = 0.1883

The 1-layer (shallow) architecture achieved the lowest MAE, suggesting that a simpler network structure may generalize better for this dataset. This result implies that the Black-Scholes function, while nonlinear, is sufficiently smooth or straightforward in its relationship that additional layers do not significantly improve predictive accuracy. Therefore, the shallow network was selected for further parameter optimization.

**2. Experimenting with Activation Functions**

To assess the effect of activation functions on model performance, we tested three functions: ReLU, Sigmoid, and Tanh.

* **Results:**
  + **ReLU:** MAE = 0.0847
  + **Sigmoid:** MAE = 0.0526
  + **Tanh:** MAE = 0.0509

The Tanh activation function achieved the lowest MAE of 0.0509, indicating it allowed the model to capture finer gradients in the nonlinear BS function. Compared to ReLU, Tanh's ability to cover both positive and negative values makes it better suited for capturing symmetrical data patterns. Thus, we selected Tanh for further tuning, as its nonlinear characteristics best matched the distribution and symmetry of the BS function.

**3. Testing the Number of Neurons**

We experimented with different numbers of neurons: 10, 50, and 100, while keeping other parameters constant.

* **Results:**
  + **10 neurons:** MAE = 0.0693
  + **50 neurons:** MAE = 0.0158
  + **100 neurons:** MAE = 0.0182

Using 50 neurons yielded the best performance, with an MAE of 0.0158. This indicates that 50 neurons provided a balance between underfitting and overfitting, adequately capturing the data's complexity without excessive redundancy. The increase to 100 neurons did not improve performance, suggesting diminishing returns, as the model was potentially capturing noise rather than the signal at this point.

**4. Optimizer Selection**

We evaluated three optimizers: Adam, RMSprop, and SGD with Momentum.

* **Results:**
  + **Adam:** MAE = 0.0158
  + **RMSprop:** MAE = 0.0750
  + **SGD with Momentum:** MAE = 0.8753

Adam achieved the lowest MAE, showing it effectively minimized the loss function. This may be attributed to Adam’s adaptive learning rate, which handles sparse gradients and noise more effectively than RMSprop or SGD with Momentum. As a result, Adam was chosen to further accelerate convergence while maintaining accuracy.

**5. Epoch Tuning**

The final step was to determine the optimal number of epochs: 100, 500, and 1000.

* **Results:**
  + **100 epochs:** MAE = 0.0205
  + **500 epochs:** MAE = 0.0099
  + **1000 epochs:** MAE = 0.0084

Training the model for 1000 epochs led to the lowest MAE, providing the best convergence. The gradual decline in error suggests that longer training better captured the relationship within the data without overfitting. This points to a sufficiently complex dataset where longer exposure aids the model in identifying nuanced relationships in the BS function.

**Summary of Optimal Parameters from Part A**

The following configuration yielded the best performance:

* **Layers:** 1
* **Activation:** Tanh
* **Neurons:** 50
* **Optimizer:** Adam
* **Epochs:** 1000

**Part B: Best Model Performance and Analysis**

Using the optimal parameters identified in Part A, we trained the model and evaluated its performance in terms of Mean Squared Error (MSE) on the test set.

* **Test Set MSE:** 0.0000996

**Convergence Analysis**  
The model's convergence over 1000 epochs was plotted for both training and validation losses. The graph showed a steady decline, indicating effective learning of the underlying pattern without overfitting. The low MSE value implies a high approximation quality for the BS function, making it suitable for real-world financial prediction tasks.

**Parameter Effects Summary**

* **Layer Depth:** A single-layer network performed better than deeper architectures, likely due to the relatively simple relationship in the BS formula.
* **Activation Function:** Tanh provided smooth learning across a continuous range, enhancing convergence speed.
* **Neuron Count:** Increasing neurons to 50 improved the model's ability to capture the function's complexity without overfitting.
* **Optimizer:** Adam yielded faster convergence compared to RMSprop and SGD with Momentum.
* **Epochs:** Extending training to 1000 epochs minimized the test MSE effectively, confirming that the model learned efficiently with more exposure.

**Part C: Random Data Split Analysis**

**Objective:** Re-evaluate the best model by randomly splitting the dataset into training, validation, and test sets to assess model robustness.

**Process**  
The data was divided into 70% training, 15% validation, and 15% testing. The model was trained using the same parameters identified in Part B.

* **Test Set MSE with Random Split:** 0.0001350

**Comparison with Part B Results**  
The MSE with the random split was slightly higher than the fixed split in Part B (0.0001350 vs. 0.0000996). This marginal increase suggests minor sensitivity to data partitioning, indicating that while the model is robust, some consistency in data ordering may slightly benefit performance. Nevertheless, the small MSE difference demonstrates that the model generalizes effectively across different splits.

**Conclusion**  
The neural network model developed in this project effectively approximated the Black-Scholes function, achieving low MSE values and consistent performance across varied data splits. The optimized configuration—a single-layer model with Tanh activation, 50 neurons, the Adam optimizer, and 1000 epochs—provided an accurate and computationally efficient solution.

**Appendix:**









