**Competition Solution Writeup: Strike-Optimized Convolutional Imputer (SOCI)**

**Overview**  
The solution employs a hybrid neural-network/matrix-completion framework specifically designed for volatility surface imputation. The model combines convolutional processing of strike relationships with regularized matrix factorization, achieving **0.00000196 MSE** on private test data through three key innovations.

**Key Components**

1. **Convolutional Strike Relationship Network**
   * **Architecture**: Implements parallel 1D convolutions across call/put strikes to capture:
     + *Local volatility smile patterns* via kernel windows spanning 5 adjacent strikes
     + *Cross-strike arbitrage constraints* through shared channel weights
   * **Residual Learning**: Uses skip connections between initial and final convolution layers to preserve moneyness structure while learning residual adjustments.
2. **Adaptive Weighted Loss Function**
   * Prioritizes accurate imputation for at-the-money (ATM) options (highest liquidity)
   * Gradually reduces emphasis on deep in/out-of-the-money strikes
3. **Bayesian-Optimized Hyperparameters**
   * The parameters for model architecture are derived using ‘optuna’ for hyperparameter tuning. (see cnn\_hypertuning.ipynb for the code)

**Training Protocol**

* **Input**: Normalized IV matrix + 20% random masking for robustness
* **Optimization**:
  + Adam with ReduceLROnPlateau (factor=0.5 after 100 loss plateaus). To reduce learning rates by a factor of 0.5 on plateaus.
  + Early stopping (200 epochs patience)

**Post-Processing**

1. **Surface Sanity Checks**:
   * Enforce put-call parity within 0.5 IV points
   * Clip predictions to [0.01, 1.5] IV range

**Steps for recreation**

1. One-to-one recreation of the leaderboard submission is quite difficult.
2. Running the ‘cnn-hypertuned\_final.ipynb’ notebook on Kaggle using cpu as hardware should produce equivalent submission with MSE score from 0.000001965 to 0.000002155.

**Attached files:**

1. cnn\_hypertuned\_final.ipynb : primary submission notebook
2. cnn\_hypertuning.ipynb : code used for hyperparameter tuning of CNN