Deep Learning Regression with Fully Homomorphic Encryption (FHE)

Overview

This project implements a deep learning-based regression model trained on a dataset, followed by secure inference using Fully Homomorphic Encryption (FHE) with TenSEAL. The key steps include:

- Training a deep learning model using TensorFlow
- Saving the trained model
- Encrypting input data using TenSEAL
- Performing inference on encrypted data
- Decrypting and displaying the predictions

Requirements

Ensure you have the following dependencies installed:

pip install numpy pandas tensorflow tenseal scikit-learn

Dataset

The dataset consists of CSV files (X_train.csv, y_train.csv) containing input features and target values. The dataset is split into 80% training and 20% validation.

Model Training

1. Load and preprocess the dataset
    - Standardize input features using StandardScaler
    - Normalize target values using MinMaxScaler
2. Define a deep learning model
    - 256, 128, and 64 neuron dense layers with ReLU activation
    - Dropout layers for regularization
    - Batch normalization for stable training
    - Adam optimizer with a learning rate of 0.0005
3. Train the model
    - 100 epochs, batch size of 32
    - Validation on the 20% held-out dataset
    - Save the trained model

Model Evaluation

After training, the model is evaluated on the validation dataset using:

- Mean Squared Error (MSE): Measures average squared errors
- Mean Absolute Error (MAE): Measures absolute differences
- R^2 Score: Measures model accuracy

Fully Homomorphic Encryption (FHE) with TenSEAL

1. Load and preprocess test data
    - Standardize input features (same as training)
    - Select a subset (e.g., 50 samples) for inference
2. Initialize TenSEAL encryption context
    - CKKS scheme with poly_modulus_degree=8192
    - coeff_mod_bit_sizes optimized for FHE operations
    - Generate Galois keys for vector operations
3. Encrypt input data
    - Convert test samples to CKKS encrypted vectors
4. Perform encrypted inference
    - Decrypt inputs
    - Predict using the trained model
    - Encrypt results
5. Decrypt and display results
    - Convert encrypted predictions back to plaintext
    - Compute average execution time per sample

Conclusion

This project successfully trains a deep learning regression model and applies FHE-based inference using TenSEAL. The approach ensures data privacy while achieving high model accuracy.