Genomic Data Classification Using a Fully Connected Neural Network

Project Overview

This project focuses on building and training a fully connected neural network from scratch to classify individuals based on their genomic SNP (Single Nucleotide Polymorphism) data. The main objective was to distinguish between different population groups using genetic information.

Key Features of the Project

- Implementation of a custom NeuralNetwork class in Python.
- Input data consists of SNP features, with each SNP encoded as 0, 1, or 2.
- Architecture includes one hidden layer with sigmoid activation functions.
- Core components such as feedforward computation, backpropagation, and weight updates were implemented manually using gradient descent.
- The model was trained using Mean Squared Error (MSE) as the loss function.
- A loss tracking vector was used to visualize convergence during training.
- Model performance was evaluated using the Matthews Correlation Coefficient (MCC) on a held-out test set.

Data Description

- Genotype matrix (geno): Contains SNP markers for each individual.
- Metadata (ind): Includes ID, gender, and population group.
- The analysis focused on specific populations such as French, Turkish, Spanish, and Unknown.

Tools and Technologies

- Programming language: Python
- Libraries: NumPy, pandas
- Environment: Google Colab
- File management: Google Drive integration for data access

Results

The loss curve showed a sharp decrease in the early training iterations, followed by stabilization around 0.1. This indicates effective learning and convergence. The final model demonstrated strong predictive performance on unseen data, successfully identifying population groups based on SNP features.

As the training set size increased, the model's predictions became more accurate. This highlights the importance of having sufficient training data for the network to generalize well to new examples.

In evaluating the model's performance, we preferred to use the Matthews Correlation

Coefficient (MCC) over loss for two main reasons:
1. MCC considers all four cases of binary classification (true positives, false positives, true negatives, and false negatives), while loss measures only the average numeric difference between the true and predicted values.
2. MCC provides a more accurate and balanced view of model performance, especially in cases where the class distribution is imbalanced.

Conclusion

This project demonstrates how a neural network can be implemented from the ground up and applied to real-world genetic data. It highlights a complete machine learning workflow—from data loading and preprocessing to training, evaluation, and interpretation of results. The model's success, particularly when evaluated using MCC, shows the potential of neural networks to uncover complex patterns in genomic data and make accurate population-level predictions.