MLP Implementation with pytorch

In this project, an MLP model was developed to classify images in the CIFAR100 dataset. The model consists of four fully connected layers with ReLU activation and a final output layer with log-softmax activation. The model was trained using the Adam optimizer and the negative log-likelihood loss function.

The initial model had a low performance with an F1 score of 0.151 on the training set and 0.146 on the test set. To improve the performance of the model, several changes can be made:

Increase the complexity of the model by adding more layers and/or more neurons to each layer. This can help the model capture more complex patterns in the data.

Tune the hyperparameters of the model such as learning rate, batch size, and regularization strength. This can help the model converge faster and avoid overfitting.

Use transfer learning by fine-tuning a pre-trained model such as ResNet or VGG. This can help the model leverage the knowledge learned from a large dataset to improve performance on a smaller dataset like CIFAR100.

Overall, it's important to try different strategies and experiment with different combinations of hyperparameters to find the best performing model.