Project Documentation

ECG Image Classification with ResNet18-

1. Overview

This project implements an end-to-end ECG image classification pipeline using deep learning. It classifies ECG images into two categories: 'Myocardial Infarction' and 'Normal'. The pipeline includes model training using ResNet18, and a Streamlit web application for real-time inference.

2. Setup and Dependencies

Required Python libraries:

- - torch
  - torchvision
  - numpy
  - pandas
  - matplotlib
  - streamlit
  - PIL

3. Data Loading and Preprocessing

The ECG images are loaded using torchvision's ImageFolder API from a structured dataset. Training data undergoes various augmentations such as resizing, flipping, rotation, affine transformation, and color jittering. Validation and test sets are resized and converted to tensors without augmentation.

4. Model Architecture

A custom model based on ResNet18 is used. The final fully connected layer is replaced to match the number of output classes (2). A dropout layer is added before the classifier to improve generalization.

5. Training and Validation

The model is trained using cross-entropy loss and the Adam optimizer. Metrics such as training and validation accuracy are tracked. The best model is saved and evaluated on the test set.

6. Testing and Evaluation

Model performance is evaluated on a separate test set. Key metrics include:
- Accuracy
- Confusion Matrix
- Precision, Recall, F1-Score

7. Web Application Interface

The project includes a Streamlit-based web application that allows users to upload ECG images and receive real-time predictions. The app uses the trained ResNet18 model and provides a simple interface with:
- Image upload feature
- Display of prediction and class description
- Streamlit caching for optimized model loading

8. Results and Insights

The model achieved good classification accuracy distinguishing between 'Myocardial Infarction' and 'Normal' classes. Using data augmentation and dropout helped improve generalization. The web app enables interactive usage, demonstrating the model's practical deployment capability.

9. Conclusion

This project showcases the use of transfer learning with ResNet18 for ECG image classification and deploys the model via a web application. Future work could include model improvement with larger datasets, interpretability via Grad-CAM, and deployment on cloud platforms.

10. References

- PyTorch: https://pytorch.org/

- ResNet Paper: https://arxiv.org/abs/1512.03385

- Streamlit: https://streamlit.io/