Self-Supervised Learning for ECG-based Emotion Recognition

1 Implementation Details

The original implementation by authors assumed a preprocessed dataset due to which I could not get comparable results to the author's implementation. I had to use a different incomplete implementation which used a library called ray - tune which had many deprecated parts of code. So I was not able to fully run the implementation.

1.1 Network Architecture

1.1.1 Signal Transformation Recognition Network

The network architecture consists of three convolutional blocks and two dense layers. The convolutional layers are shared among tasks, while the dense layers are specific to each task. Each convolutional block includes:

- Two 1D convolutional layers with ReLU activation.
- A max-pooling layer (size: 8).
- The number of filters increases from 32 to 128 across blocks.
- Kernel sizes reduce from 32 to 8 after each block.

The final layers consist of global max pooling, followed by task-specific dense layers. The dense layers use dropout (60%) and L2 regularization (with $\beta = 0.0001$).

1.1.2 Emotion Recognition Network

This network reuses the convolutional layers of the signal transformation recognition network. Two dense layers with 64 hidden units and a sigmoid activation follow. The transferred convolutional layers are frozen, and only the dense layers are trained using the ECG signals and emotion labels.
A Self-supervised learning framework was employed, using ECG data to pre-train a convolutional neural network (CNN). The model consists of convolutional layers followed by batch normalization and ReLU activation. The architecture also includes a fully connected layer for emotion classification.

1.2 Pretext Task

For self-supervision, a contrastive learning approach was used. The pretext task was defined by generating augmented views of the ECG signals

1.3 Fine-tuning

After the pre-training, the model was fine-tuned on a labeled dataset for emotion recognition using categorical cross-entropy loss and stochastic gradient descent (SGD) optimization.

2 Dataset Description

The dataset used for this project is the *DREAMER* dataset which is a multi-modal database comprising of electrocardiogram (ECG) signals with **Sampling Rate:** 256 Hz from 23 participants recorded during audio-visual stimuli for affect elicitation. Participants self-assessed their emotions in terms of valence, arousal, and dominance.

2.1 Dataset Processing

The *DREAMER* dataset was originally in *.mat* format and was processed into *.npy* files for easier handling. ECG baseline and stimuli data were extracted, **normalized** using calculated mean and standard deviation, and segmented into windows of 2560 samples. Labels for valence, arousal, and dominance were also generated for each window, and the final processed data was saved in *.npy* format for efficient model training.

3 Results

```
Trial train_pretext_full_config_ce31a_00001 started with configuration:
 Trial train_pretext_full_config_ce31a_00001 config
 pretext/adam/lr                              0.0024396003748203415
 pretext/batch_size                                              32
 pretext/scheduler/cycles                                         2
 pretext/scheduler/decay                         0.9513865638735618
 pretext/scheduler/type                                        none
 pretext/scheduler/warmup                                         5

Trial train_pretext_full_config_ce31a_00000 started with configuration:
 Trial train_pretext_full_config_ce31a_00000 config
 pretext/adam/lr                              0.0020220503805149607
 pretext/batch_size                                              64
 pretext/scheduler/cycles                                         1
 pretext/scheduler/decay                         0.9671129657621143
 pretext/scheduler/type                             cosine_w_restarts
 pretext/scheduler/warmup                                         8
```

Figure 1: Training Config

```
) Epoch: 71       Training Loss: 0.057726        Validation Loss: 0.045543 [repeated 2x across cluster]
)                 Training Accuracy: 0.778       Validation Accuracy: 0.790 [repeated 2x across cluster]
) lr for epoch 72  0.0002698912820721283 [repeated 2x across cluster]
) Epoch: 53       Training Loss: 0.089028        Validation Loss: 0.072531
)                 Training Accuracy: 0.475       Validation Accuracy: 0.504
) lr for epoch 54  0.0024396003748203415
```

Figure 2: Intermediate Accuracy

```
239) Epoch: 99        Training Loss: 0.056385         Validation Loss: 0.043842
239)                  Training Accuracy: 0.789        Validation Accuracy: 0.801
238) Epoch: 74        Training Loss: 0.130488         Validation Loss: 0.122613
238)                  Training Accuracy: 0.141        Validation Accuracy: 0.142
238) lr for epoch 75  0.0024396003748203415
```

Figure 3: Final Accuracy