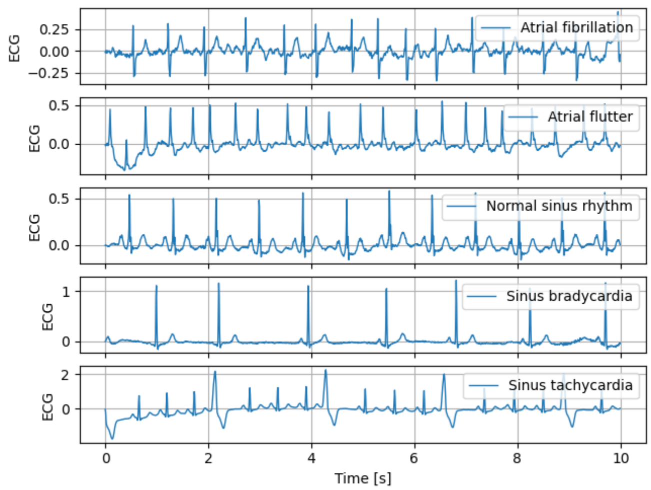
Neural Network Lab 2

Exercise 1: ECG Rhythm Classification

Question 1. Visually, what are the differences between the different rhythms?

The main differences between the rhythms are the heart rate and the beat regularity.

- **Normal sinus rhythm**: regular RR intervals, with clear and consistent P-QRS-T morphology.
- **Sinus bradycardia**: P-wave before QRS complex, lower heart rate (<60 bpm at rest), with longer but regular RR intervals.
- **Sinus tachycardia**: P-wave before QRS complex, higher heart rate (>100 bpm at rest), with shorter but regular RR intervals.
- **Atrial fibrillation**: no distinct P-wave before QRS complex, very irregular RR intervals, higher heart rate
- **Atrial flutter**: sawtoothed patterned of P-wave, irregular RR intervals, higher heart rate.



Question 2. Comment the metrics shown in TensorBoard and the confusion matrices. Does the model overfit? Are there some rhythms that are difficult to classify?

There is very **limited overfitting** as shown by the small difference in terms of accuracy for the training and validation sets. The performance **does not reach a plateau**. It could therefore be improved by training the model for more epochs, while being careful not to overfit. Both the training and validation losses would probably decrease further.

Based on the confusion matrices, it seems that **normal sinus rhythm, sinus bradycardia, and sinus tachycardia are classified correctly** in most cases. However, the model has some trouble discriminating between **atrial fibrillation and atrial flutter**. This could be caused by similarities observed in question 1.

Question 3. Define two custom models to classify cardiac rhythms from ECG signals.

No generic answer since the models must be defined by the students.

Question 4. How do the two custom models perform? Do they overfit? Do they outperform the first model?

No generic answer since the models must be defined by the students.

Exercise 2: Heart Rate Estimation

Question 1. The windows are scaled but not centered. Why?

As mentioned in the notebook, the signals are **already band-pass filtered** between 0.4 and 4 Hertz. This step **removes the baseline** wander. Therefore, the signals already have mean near zero. The scaling to unit variance facilitates the training of the model.

Question 2. Based on metrics shown in TensorBoard, does using the acceleration signal in addition to the PPG signal help to improve performance?

Using the acceleration signals in addition to the PPG signal seems to be important as the **MSE and MAE are lower** for the CNN model using the acceleration on both the training and validation sets. It suggests that acceleration signals provide complementary information.

Question 3. What can you say about the predictions computed with the two models?

The heart rate predicted by the **CNN** model that uses the **acceleration** signal is not perfect, but it is relatively **close to the reference**. The acceleration signal probably helps account for **movement artifacts**. By contrast, the predictions obtained with the CNN model **without the acceleration** signal are completely wrong, with sometimes **non-physiological** values (above 200bpm at rest and above 500bpm when walking). This model failed to learn how to predict heart rate.

Question 4. What is the effect of batch normalization on the training procedure and on the performance metrics?

Batch normalization improves both training process and final performance by **reducing internal covariate shift**. It makes training **faster and more stable** (the loss decreases faster with batch normalization than without). In addition, it also helps to achieve **lower MSE and MAE** on the validation set.

Question 5. Visually is there a large difference between the CNN models with and without batch normalization on these examples?

In these two examples, there are few differences between the CNN models trained with and without batch normalization. Both models appear to be **slightly biased** but the predictions obtained with batch normalization seem **less noisy**, especially when walking.

Question 6. Does using dropout help to reduce overfitting compared to the same model without dropout?

Dropout randomly removes some nodes of the network at every training iteration, which prevents relying too much on particular nodes. There is **almost no difference in terms of MSE and MAE** between the CNN models trained with and without dropout on the **validation** set. However, both the MSE and MAE are **higher** for the model trained with dropout on the **training** set. This shows that dropout is effective to **reduce overfitting**.

Question 7. What is the best model in terms of MSE and MAE? What can you say about batch normalization and dropout?

The two best CNN models are the ones trained with batch normalization (with or without dropout). Dropout helps to reduce overfitting, but it is not clear if it helps to improve overall performance. In fact, the model trained without dropout is the best on the validation set but the one trained with dropout is the best on the test set. However, this might be an artifact of the small dataset (it includes only 15 subjects).