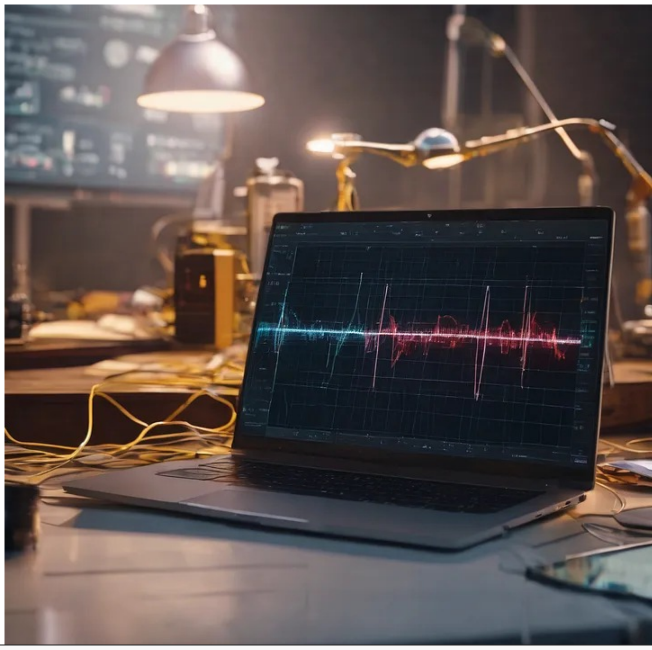
Navigating the Signal: ECG Data Filter Algorithms

Sumeyya Eser · Follow

3 min read · Feb 15, 2024



algorithms and their Python implementations.

1. Linear Filters

Linear filters modify signals based on a linear combination of input samples. We'll implement a basic low-pass filter using the convolution operation.

```
import numpy as np
import matplotlib.pyplot as plt

def low_pass_filter(signal, cutoff_freq, sampling_freq):
    # Designing filter kernel
    t = np.arange(-10, 10, 1/sampling_freq)
    kernel = np.sinc(2 * cutoff_freq * t)
    kernel /= np.sum(kernel)

    # Convolve signal with filter kernel
    filtered_signal = np.convolve(signal, kernel, mode='same')
    return filtered_signal

# Sample ECG signal (replace with actual data)
sampling_freq = 1000  # Hz
t = np.arange(0, 10, 1/sampling_freq)
ecg_signal = np.sin(2 * np.pi * 10 * t) + 0.5 * np.sin(2 * np.pi * 20 * t)
```

```
# Apply low-pass filter
cutoff_freq = 15  # Hz
filtered_ecg = low_pass_filter(ecg_signal, cutoff_freq, sampling_freq)

# Plot original and filtered signals
plt.figure(figsize=(10, 5))
plt.plot(t, ecg_signal, label='Original ECG')
plt.plot(t, filtered_ecg, label='Filtered ECG')
plt.xlabel('Time (s)')
plt.ylabel('Amplitude')
plt.title('ECG Signal with Low-pass Filtering')
plt.legend()
plt.grid(True)
plt.show()
```

2. Adaptive Filters
Adaptive filters adjust parameters based on input signals, making them effective in non-stationary environments. Let's implement a basic LMS adaptive filter to remove noise.

```
from scipy.signal import lfilter

def adaptive_filter(signal, noise, mu=0.01, order=4):
    filtered_signal, _, _ = lfilter([mu] * order, 1, signal, zi=noise)
    return filtered_signal

# Generate noisy ECG signal (replace with actual data)
noise = np.random.normal(0, 0.5, len(ecg_signal))

# Apply adaptive filtering
filtered_ecg_adaptive = adaptive_filter(ecg_signal, noise)

# Plot original and adaptively filtered signals
plt.figure(figsize=(10, 5))
plt.plot(t, ecg_signal, label='Original ECG')
plt.plot(t, filtered_ecg_adaptive, label='Adaptive Filtered ECG')
plt.xlabel('Time (s)')
plt.ylabel('Amplitude')
plt.title('ECG Signal with Adaptive Filtering')
plt.legend()
plt.grid(True)
plt.show()
```

3. Wavelet Transform-Based Filters
Wavelet transforms decompose signals into frequency components, aiding noise removal while preserving features. Let's use the PyWavelets library for wavelet denoising.

```
import pywt

def wavelet_denoise(signal, wavelet='db4', level=3):
    coeffs = pywt.wavedec(signal, wavelet, level=level)
    threshold = np.sqrt(2 * np.log(len(signal)))
    coeffs = [pywt.threshold(c, threshold, mode='soft') for c in coeffs]
    denoised_signal = pywt.waverec(coeffs, wavelet)
    return denoised_signal

# Apply wavelet denoising
denoised_ecg_wavelet = wavelet_denoise(ecg_signal)

# Plot original and denoised signals
plt.figure(figsize=(10, 5))
plt.plot(t, ecg_signal, label='Original ECG')
plt.plot(t, denoised_ecg_wavelet, label='Denoised ECG (Wavelet)')
plt.xlabel('Time (s)')
plt.ylabel('Amplitude')
plt.title('ECG Signal with Wavelet Denoising')
plt.legend()
plt.grid(True)
plt.show()
```

Python provides powerful tools for implementing and visualizing ECG data filtering algorithms. By applying these techniques, healthcare professionals can enhance diagnostic accuracy and improve patient care outcomes in cardiac monitoring and analysis. Experimenting with various algorithms and parameters can further refine the filtering process, ensuring optimal signal quality for clinical interpretation.

Ecg Filtering Algorithms

Written by Sumeyya Eser

Follow

0 Followers

Software Developer

More from Sumeyya Eser

Sumeyya Eser

Fetch and Cache Mechanism with Couchbase Lite DB in React Nativ...

Minimizing the data usage of React Native apps is of great importance in terms of user...

4 min read · Nov 13, 2023

Sumeyya Eser

Yurt Dışında Staj Nasıl Yapılır? Baştan Sona Bütün Süreç

Kariyer yolculuğunda kendimize yapacağımız en büyük katkılardan biri olan yurtdışı stajını...

4 min read · Sep 1, 2022

See all from Sumeyya Eser

Recommended from Medium

Zippo

List Comprehensions for Dummies

At least once in a lifetime, any of us would read some Python code between two squar...

· 4 min read · 6 days ago

133

Juan C Olamendy

Deep TDA. A new dimensionality reduction algorithm

Introduction

3 min read · Oct 19, 2023

48

Lists

Practical Guides to Machine Learning

10 stories · 1206 saves

Staff Picks

602 stories · 827 saves

General Coding Knowledge

20 stories · 1021 saves

Thomas Smith in The Generator

Kate Middleton's Doctored Photo is a Sign of Something Much More...

Trust in photos is eroding fast, and tech is to blame

· 6 min read · 4 days ago

7.4K 140

Tim Berners-Lee

Marking the Web's 35th Birthday: An Open Letter

Sir Tim Berners-Lee's open letter to mark the occasion of the Web's 35th Birthday.

5 min read · 4 days ago

7.3K 102

Hazal Gültekin

Part2: Demand Forecast Modeling

It is the continuation of 👉Part1: Demand Forecast Modeling.

17 min read · 4 days ago

53 1

Samir Saci in Towards Data Science

Data Science for Sustainability- Green Inventory Management

Simulate the impact of store delivery frequency on the CO2 emissions of a fashion...

10 min read · 2 days ago

66

See more recommendations