

1. Цель лабораторной работы

классифицировать временные ряды электроэнцефалографии для определения эпилепсии

2. введение

В ходе данного эксперимента рассматриваются несколько взаимосвязанных тем:

1. Построение обучающего набора данных

- Использовать открытый набор ЭЭГ-данных PhysioNet (с разметкой припадков):
<https://physionet.org/content/chbmit/1.0.0/chb08/#files-panel>
- Выбрать не менее 50 фрагментов с припадками и не менее 50 фрагментов без припадков.

2. Предобработка данных

- Вырезать из выбранных записей подходящие временные сегменты.
- Преобразовать каждый сегмент ЭЭГ в вейвлет-спектрограмму (временно-частотное изображение); можно разделять по диапазонам ритмов или генерировать общую вейвлет-карту.
- В итоге должно получиться 100 изображений:
 - 50 изображений с припадками → поместить в отдельную папку **A**;
 - 50 изображений без припадков → поместить в отдельную папку **B**.

3. Построение модели

- На основе полученных изображений спроектировать и обучить сверточную нейронную сеть для бинарной классификации наличия/отсутствия эпилептического припадка на ЭЭГ.

- После обучения сохранить файл модели и проверить её эффективность на тестовом наборе.

Проверка среды выполнения

```
# фильтр Баттерфорда для вырезания полосы частот
from scipy import signal
def butter_bandpass_filter(data, lowcut, highcut, fs, order=5):
    nyquist = 0.5 * fs
    low = lowcut / nyquist
    high = highcut / nyquist
    b, a = signal.butter(order, [low, high], btype='band')
    filtered_data = signal.lfilter(b, a, data)
    return filtered_data
```

Этот код реализует полосовой фильтр (band-pass filter) на основе фильтра Баттерворта (Butterworth filter): он сохраняет компоненты сигнала в заданном диапазоне частот (от lowcut до highcut) и подавляет остальные частоты.

Эта код реализует的是一个带通滤波器 (bandpass filter)，使用的是巴特沃斯滤波器 (Butterworth filter)，用于保留信号中某一指定频率范围 (lowcut 到 highcut) 之间的成分，同时滤除其他频率部分。

```
# функция преобразования Гильберта для построения огибающей
from scipy.fft import fft, ifft
def my_func_hilbert(x):
    N = len(x)
    xf = fft(x)
    xf[1:N//2] *= 2
    if N % 2 == 1:
        xf[N//2] *= 2
    xf[N//2 + 1:] = 0
    xa = ifft(xf)
    assert np.allclose(xa.real, x)
    return xa
```

Этот код вручную реализует преобразование Хилберта через преобразование Фурье и возвращает аналитический сигнал, который можно использовать для:

- извлечения огибающей сигнала (мгновенной энергии);
- расчёта мгновенной фазы и частоты;
- анализа модуляции/демодуляции сигнала.

По сравнению с `scipy.signal.hilbert` это более низкоуровневый метод, позволяющий вручную контролировать спектральные компоненты.

Эта код использует Фурье-преобразование для реализации Hilbert-преобразования, возвращая аналитический сигнал, который можно использовать для: извлечения огибающей (мгновенной энергии), расчёта мгновенной фазы и частоты, анализа модуляции/демодуляции сигнала. Это более низкоуровневый метод, чем `scipy.signal.hilbert`, позволяющий вручную контролировать спектральные компоненты.

```
# модули для обработки временных рядов (убрать тренд и аппроксимировать)
class Detrending:
    def plot(t,y,slope,intercept):
        fig = plt.figure(figsize=(10,7))
        ax = plt.subplot(111)
        plt.scatter(t,y, s=5)
        plt.plot(t, slope*t + intercept, 'r',
                 label='y={:.2f}x+{:.2f}'.format(slope,intercept))

        plt.grid(True)
        plt.legend(fontsize=12)
        plt.xlabel('time')
        plt.ylabel('Y')
        plt.show()

    def line_fit(t,y):
        from scipy import stats
        slope, intercept, r_value, p_value, std_err = stats.linregress(t,y)
        #line = slope*t + intercept
        return slope, intercept

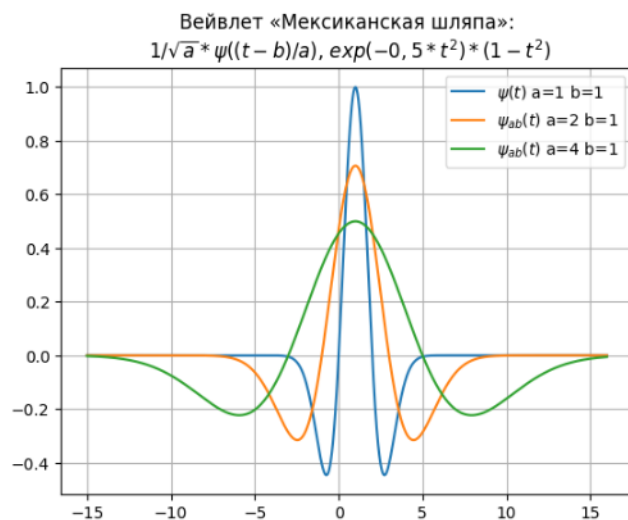
    def trend_off(t,y):
        from scipy import stats
        slope, intercept, r_value, p_value, std_err = stats.linregress(t,y)
        line = slope*t + intercept

        y_nontrend = y-line
        return y_nontrend
```

Этот фрагмент кода определяет класс **Detrending**, предназначенный для обработки временных рядов методом детренда: он удаляет из данных линейный тренд и визуализирует исходные значения вместе с их линейной аппроксимацией.

Эта код определяет класс **Detrending**, предназначенный для обработки временных рядов методом детренда: он удаляет из данных линейный тренд и визуализирует исходные значения вместе с их линейной аппроксимацией.

```
x = np.arange(-15,16,0.01)
def w(a,b,t):
    f = (1/a**0.5) * np.exp(-0.5*((t-b)/a)**2) * (1 - ((t-b)/a)**2)
    return f
plt.title("Вейвлет «Мексиканская шляпа»: \n $1/\sqrt{a}*\psi((t-b)/a)$, $exp(-0,5*t^2)*(1 - t^2)$")
b = 1
y=[w(1,b,t) for t in x]
plt.plot(x,y, label = '$\psi(t)$ a=1 b=1')
y=[w(2,b,t) for t in x]
plt.plot(x,y, label="$\psi_{ab}(t)$ a=2 b=1")
y=[w(4,b,t) for t in x]
plt.plot(x,y, label="$\psi_{ab}(t)$ a=4 b=1")
plt.legend(loc='best')
plt.grid(True)
plt.show()
```



Этот фрагмент кода демонстрирует, как меняется форма **вейвлета «мексиканская шляпа»** при разных значениях коэффициента масштабирования a :

- $a = 1$ — волна узкая;
- $a = 2$ — волна становится шире;
- $a = 4$ — волна ещё шире и более плоская.

Это наглядно показывает свойство масштабирования в непрерывном вейвлет-преобразовании. В обработке сигналов и распознавании образов операции масштабирования и сдвига вейвлета используются для извлечения локальных признаков и многомасштабного анализа.

Эта код展示了***“墨西哥帽小波”在不同缩放参数 a 下的形状变化**： $a=1$: 波形较窄； $a=2$: 波形变宽； $a=4$: 波形更宽更平。这有助于理解连续小波变换中小波的缩放性质。在信号处理和模式识别中，小波的缩放和位移操作可用于提取局部特征和多尺度分析。

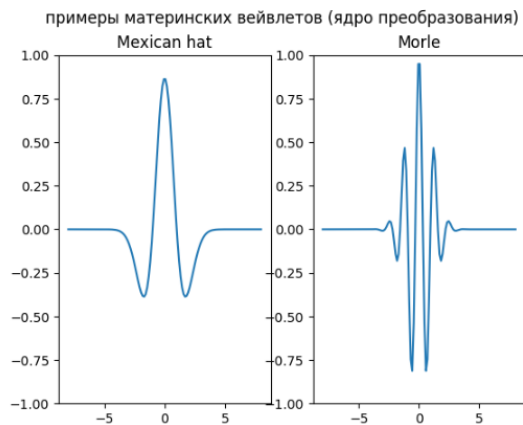
```
#discrete wavelet
#wavelet_d = pywt.Wavelet('db5')
#[phi, psi, x] = wavelet_d.wavefun(level=1)
#continuous wavelet
# The wavefun() method can be used to calculate approximations of scaling function (phi) and wavelet function (psi) at the given level of refinement.
wavelet_c = pywt.ContinuousWavelet('mexh')
[psi,x] = wavelet_c.wavefun(level = 7)
wavelet_c = pywt.ContinuousWavelet('morl')
[psi_morle,x] = wavelet_c.wavefun(level = 7)

fig, (ax1, ax2) = plt.subplots(1, 2)
fig.suptitle('примеры материнских вейвлетов (ядро преобразования)')

ax1.plot(x, psi)
ax1.set_ylim(-1, 1)
ax1.set_title('Mexican hat')
# Вейвлет «Мексиканская шляпа»: \n1/\sqrt{a}*exp(-0,5*t^2/a^2)*(t^2-1)$")

ax2.plot(x, psi_morle)
ax2.set_ylim(-1, 1)
ax2.set_title('Morle')

Text(0.5, 1.0, 'Morle')
```



На рисунке показаны формы двух «материнских» вейвлет-функций (Mexican Hat и Morlet), применяемых в непрерывном вейвлет-преобразовании:

- **Mexican Hat**

(мексиканская шляпа) — функция, близкая к гауссовой форме, хорошо подходит для обнаружения кратковременных изменений;

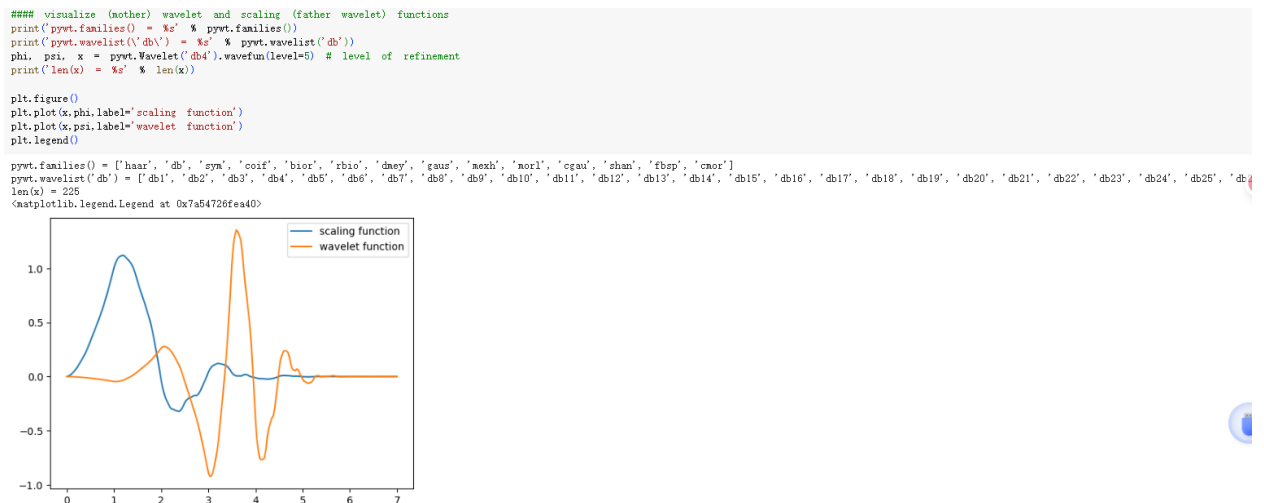
- **Morlet (Морле)** —

модулированная гауссовая волна с хорошей частотной локализацией,

- 该图展示了连续小波变换中的两种“母小波”函数（Mexican Hat 和 Morlet）的形状：Mexican Hat（墨西哥帽）：类高斯波形，适合检测瞬时变化；Morlet（莫雷）：调制的高斯波，具有良好的频率局部性，适合周期性成分分析。这些母小波是连续小波变换中用于分析信号的***“核函数”***，它们通过缩放和平移应用于目标信号，从而提取多尺度特征

удобная для анализа периодических
компонентов.

Эти материнские вейвлеты
выступают в роли «ядерных функций»
непрерывного вейвлет-преобразования;
масштабируя и сдвигая их по отношению
к анализируемому сигналу, можно
извлекать его многомасштабные
характеристики.



Пояснение результатов:

Две кривые:

- одна — сглаженная

функция масштабирования $\phi(x)$,
представляющая низкочастотное
приближение;

- другая — волновая

функция $\psi(x)$ с положительными и отрицательными колебаниями, предназначенная для извлечения детальной информации сигнала.

Функция $\phi(x)$ служит базисной функцией для построения низкочастотной (грубой) составляющей, а $\psi(x)$ — базисной функцией для построения высокочастотной (детальной) составляющей.

Обе функции лежат в основе **многоуровневого анализа (MRA)**.

Вейвлет **db4** по сравнению с **Haar** более сглаженный и лучше подходит для обработки стационарных сигналов, например при сжатии изображений, анализе ЭКГ и т. д.

Результаты:
две кривые:

одна — это гладкая scaling function $\phi(x)$, представляющая низкочастотную;

одна — это функция с положительными и отрицательными колебаниями wavelet function $\psi(x)$, используемая для извлечения информации. $\phi(x)$ — это базисная функция для построения низкочастотной (грубой) части;

$\psi(x)$ — это базисная функция для построения высокочастотной (детальной) части;

эти две функции составляют основу многоуровневого анализа (MRA);

db4 вейвлет по сравнению с haar более гладкий, подходит для обработки стационарных сигналов, например для сжатия изображений, анализа ЭКГ и т. д.


```
#!/usr/bin/env python
# -*- coding: utf-8 -*-

# Plot scaling and wavelet functions for db, sym, coif, bior and rbio families

import itertools

import matplotlib.pyplot as plt

import pywt

plot_data = [('db', (4, 3)),
              ('sym', (4, 3)),
              ('coif', (3, 2))]

for family, (rows, cols) in plot_data:
    fig = plt.figure()
    fig.subplots_adjust(hspace=0.2, wspace=0.2, bottom=.02, left=.06,
                        right=.97, top=.94)

    colors = itertools.cycle('bgrcmk')

    wnames = pywt.wavelist(family)
    i = iter(wnames)
    for col in range(cols):
        for row in range(rows):
            try:
                wavelet = pywt.Wavelet(next(i))
            except StopIteration:
                break
            phi, psi, x = wavelet.wavefun(level=5)

            color = next(colors)
            ax = fig.add_subplot(rows, 2 * cols, 1 + 2 * (col + row * cols))
            ax.set_title(wavelet.name + " phi")
            ax.plot(x, phi, color)
            ax.set_xlim(min(x), max(x))

            ax = fig.add_subplot(rows, 2*cols, 1 + 2*(col + row*cols) + 1)
            ax.set_title(wavelet.name + " psi")
            ax.plot(x, psi, color)
            ax.set_xlim(min(x), max(x))

for family, (rows, cols) in [('bior', (4, 3)), ('rbio', (4, 3))]:
    fig = plt.figure()
```

```
for family, (rows, cols) in [('bior', (4, 3)), ('rbio', (4, 3))]:
    fig = plt.figure()
    fig.subplots_adjust(hspace=0.5, wspace=0.2, bottom=.02, left=.06,
                        right=.97, top=.94)

    colors = itertools.cycle('bgrcmk')
    wnames = pywt.wavelist(family)
    i = iter(wnames)
    for col in range(cols):
        for row in range(rows):
            try:
                wavelet = pywt.Wavelet(next(i))
            except StopIteration:
                break
            phi, psi, phi_r, psi_r, x = wavelet.wavefun(level=5)
            row *= 2

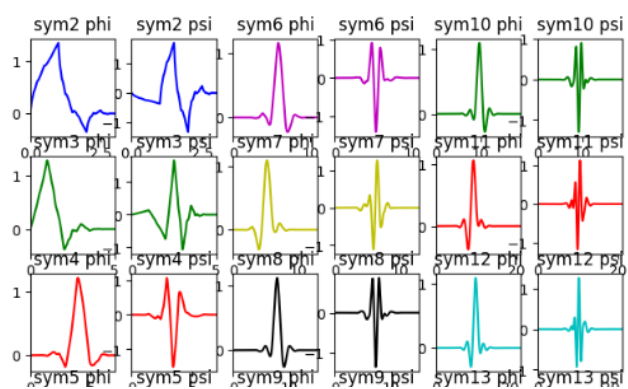
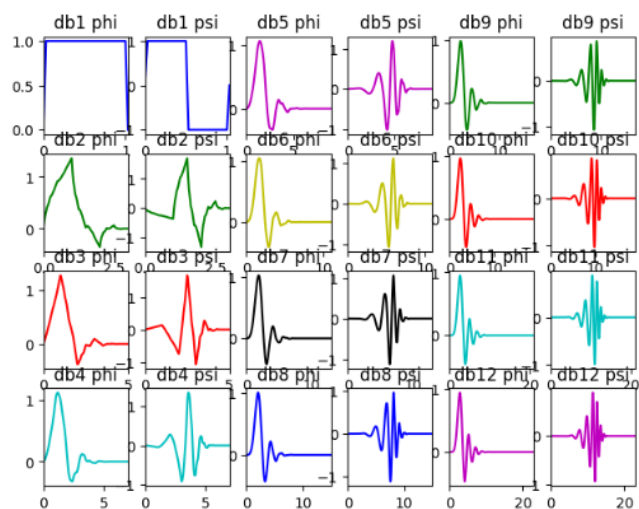
            color = next(colors)
            ax = fig.add_subplot(2*rows, 2*cols, 1 + 2*(col + row*cols))
            ax.set_title(wavelet.name + " phi")
            ax.plot(x, phi, color)
            ax.set_xlim(min(x), max(x))

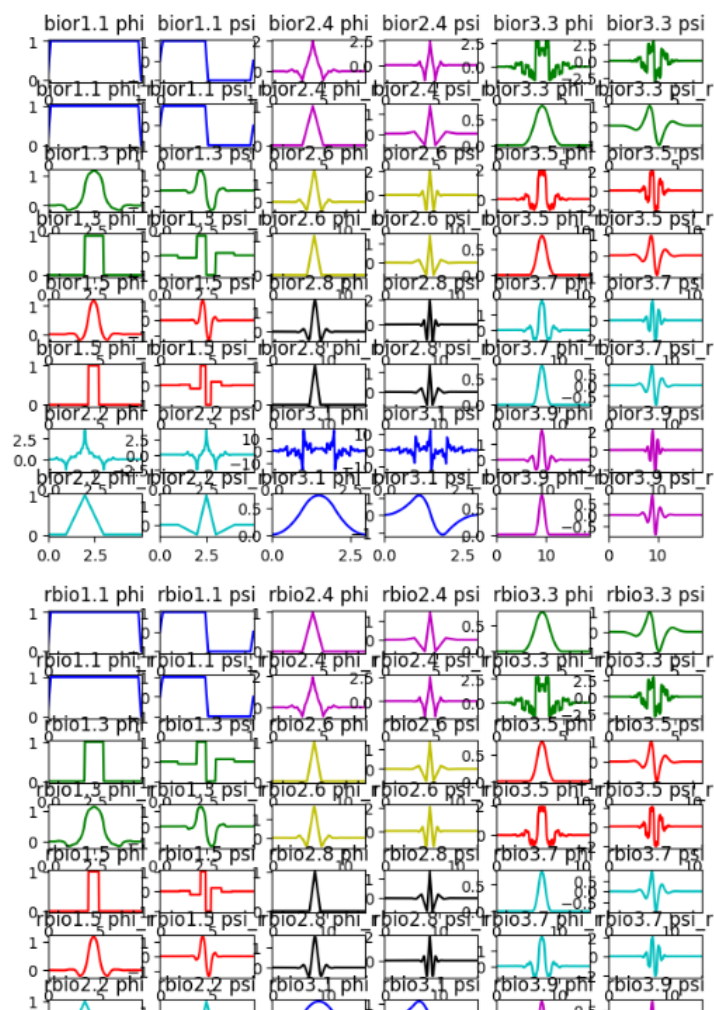
            ax = fig.add_subplot(2*rows, 2*cols, 2*(1 + col + row*cols))
            ax.set_title(wavelet.name + " psi")
            ax.plot(x, psi, color)
            ax.set_xlim(min(x), max(x))

            row += 1
            ax = fig.add_subplot(2*rows, 2*cols, 1 + 2*(col + row*cols))
            ax.set_title(wavelet.name + " phi_r")
            ax.plot(x, phi_r, color)
            ax.set_xlim(min(x), max(x))

            ax = fig.add_subplot(2*rows, 2*cols, 1 + 2*(col + row*cols) + 1)
            ax.set_title(wavelet.name + " psi_r")
            ax.plot(x, psi_r, color)
            ax.set_xlim(min(x), max(x))

plt.show()
```

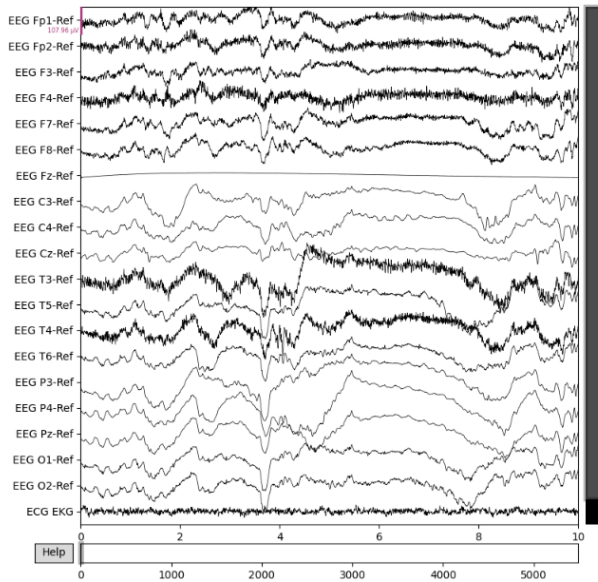




Этот полный скрипт Python предназначен для пакетного построения графиков волновых функций (ψ) и функций масштабирования (ϕ) различных семейств дискретных вейвлетов (db, sym, coif, bior, rbio). Он наглядно демонстрирует различия между этими семействами и служит отличным

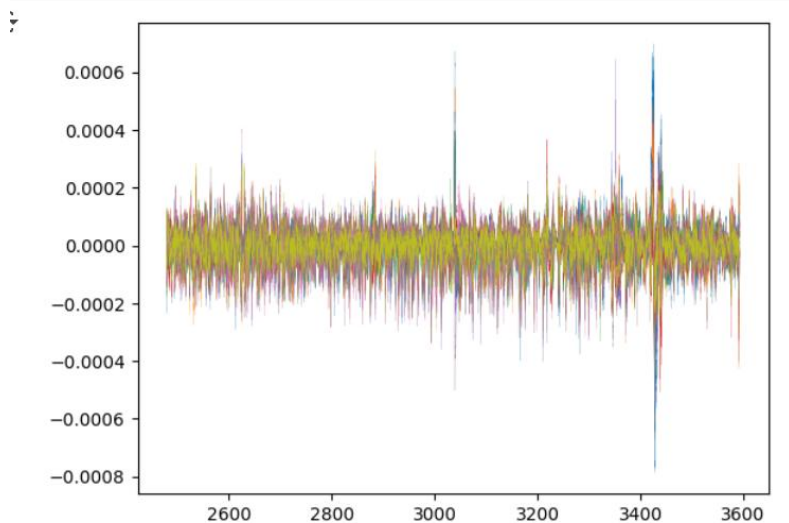
этой完整的 Python 脚本用于批量绘制多种离散小波族 (db、sym、coif、bior、rbio) 的小波函数 (ψ) 和尺度函数 (ϕ)，可视化它们在不同小波族中的表现差异，是理解小波分析中“母小波”和“父小波”形状的一个极佳工具。

инструментом для понимания
формы «материнского» и «отцовского»
вейвлетов в вейвлет-анализе



Визуализировать многоканальные
биоэлектрические сигналы из EDF-файла
и вывести список всех названий каналов,
чтобы упростить последующий выбор
объектов анализа.

可視化 EDF 文件中的多通道生物电
信号； 列出信号中包含的所有通道名
称，便于后续选择分析对象。



Каждая линия на графике соответствует одному каналу ЭЭГ.

По горизонтальной оси откладывается время (в секундах), по вертикальной — напряжение (обычно в мкВ).

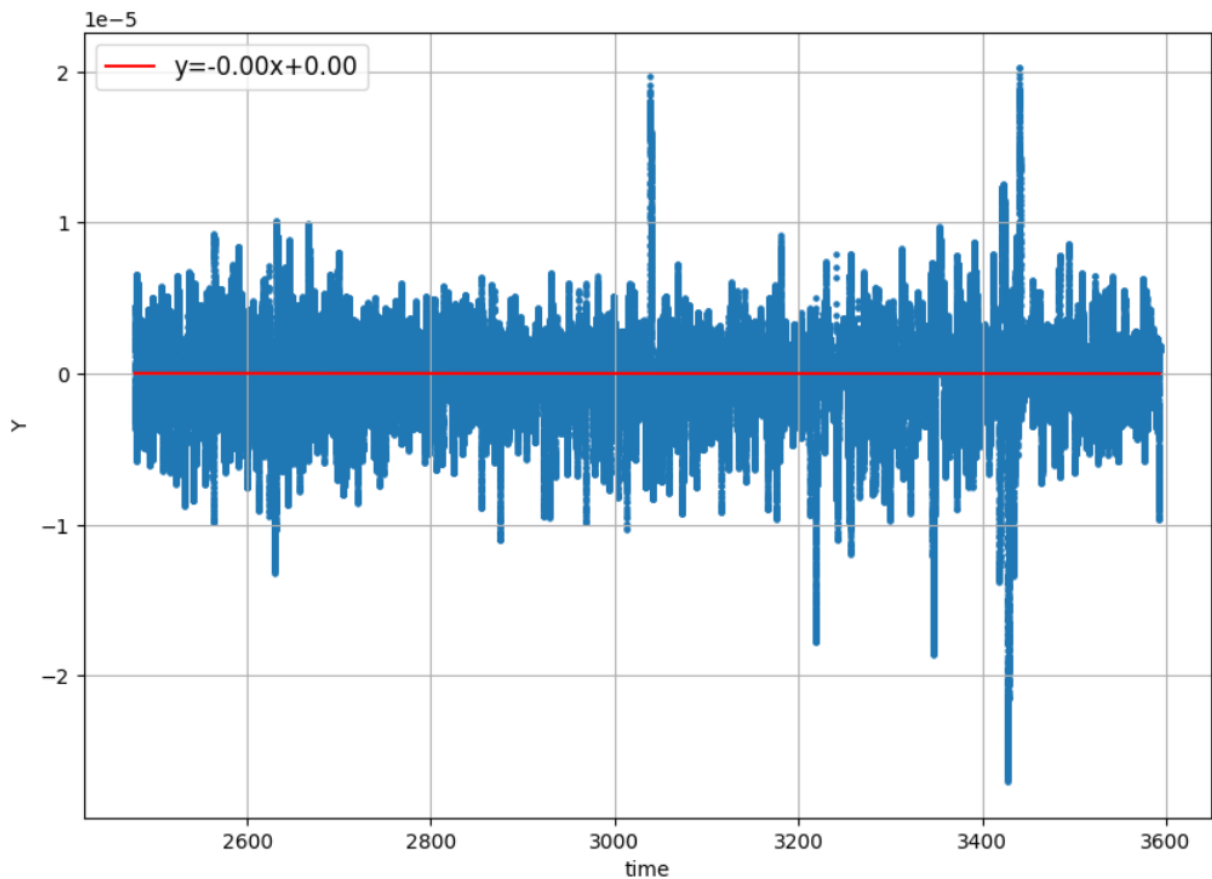
Такой график помогает определить, в каких каналах перед приступом и после него появляются аномальные колебания (например, острые волны, «спайки» и т. д.).

Если кривые получаются слишком плотными, можно добавить вертикальное смещение для каждой линии или разбить визуализацию на несколько подграфиков.

每条线代表一个 EEG 通道；横轴是时间（单位：秒），纵轴是电压值（单位通常为 μV ）；

这种图形有助于观察在癫痫发作前后，哪些通道出现了异常波动（例如尖波、棘波等）；

如果波形太密集，可以考虑添加偏移量显示每条曲线，或绘制多个子图。



Выполнить линейную
аппроксимацию (детренд) усреднённого
сигнала ЭЭГ;

Визуализировать исходный
сигнал вместе с линией тренда;

Удалить линейный тренд из
сигнала;

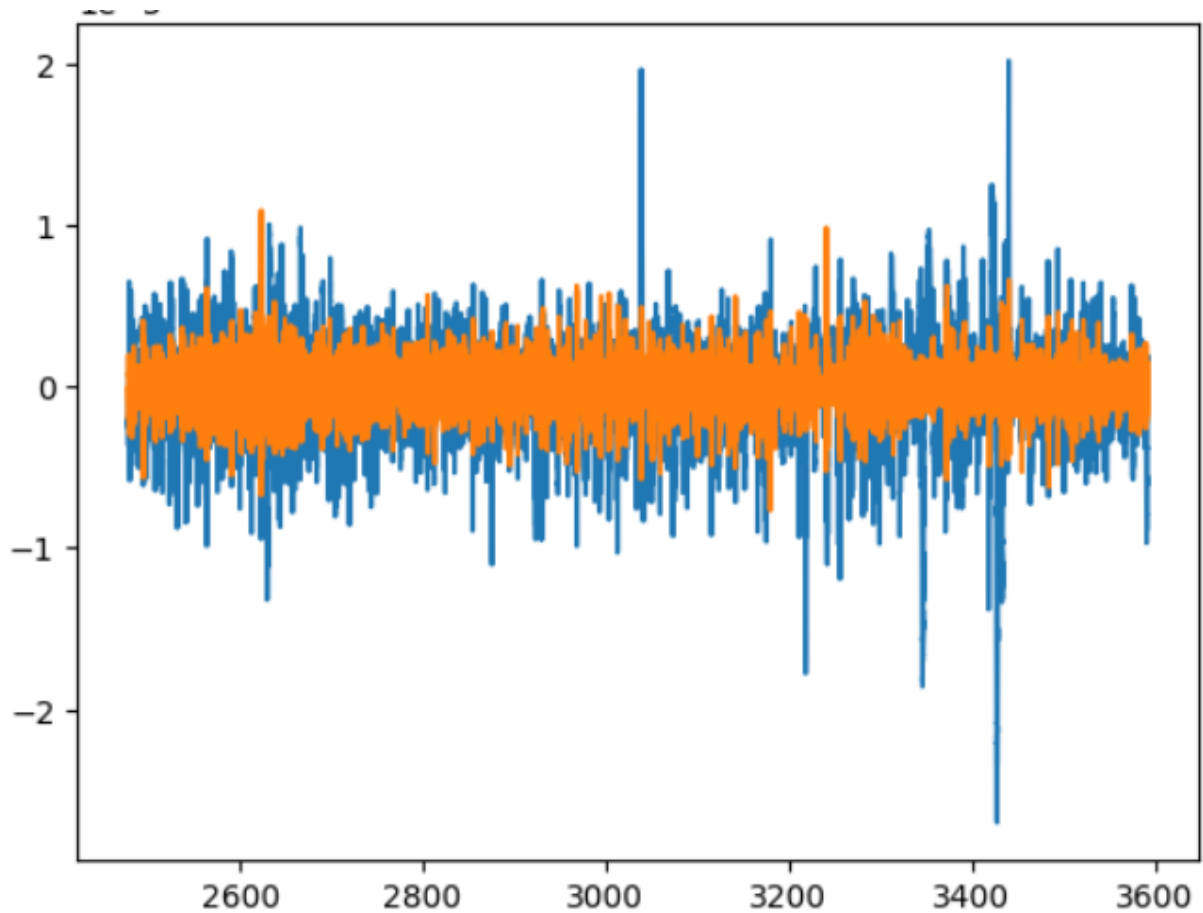
Повторно построить график
детрендрованного сигнала.

对 EEG 平均信号进行线性拟合（去
趋势）；

可视化原始信号与趋势线；

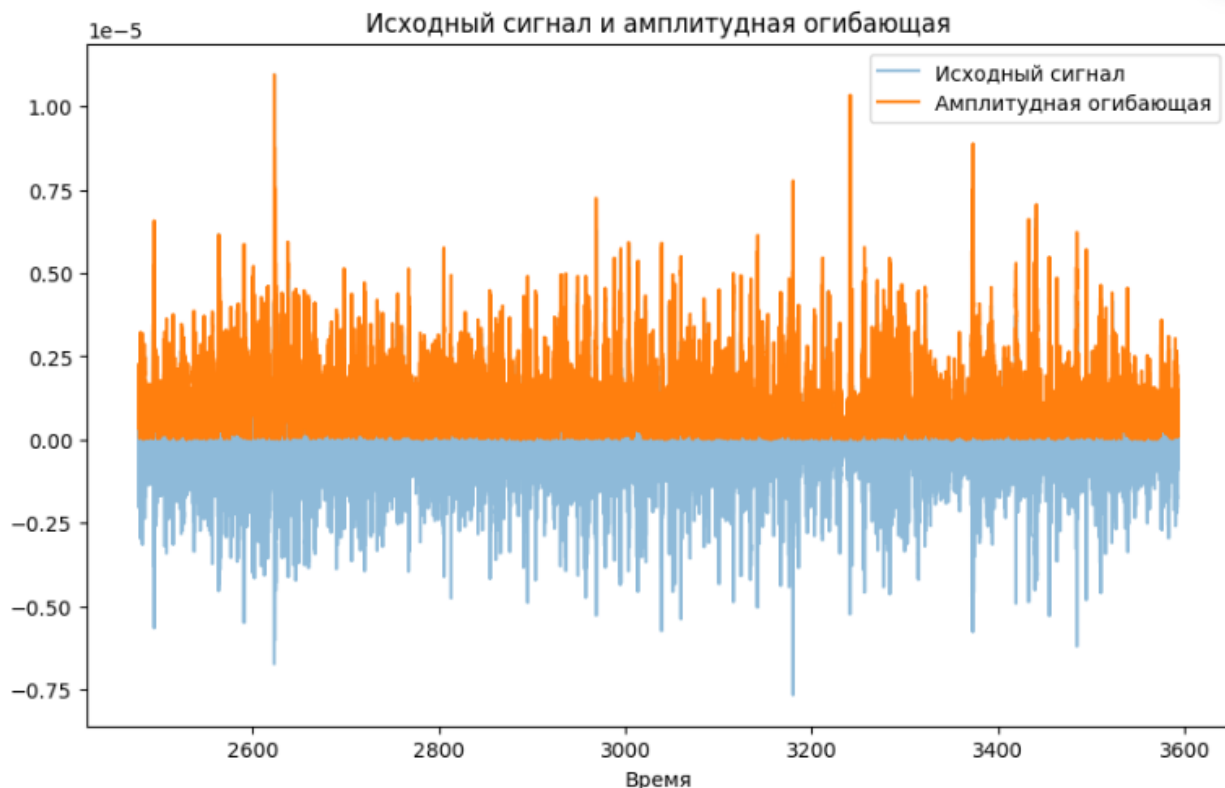
从信号中去除线性趋势；

再次绘制去趋势后的信号。



Этот фрагмент кода выполняет
полосовую фильтрацию (band-pass filter)
ЭЭГ-сигнала, что обычно используется
для удаления низкочастотного дрейфа и
высокочастотного шума.

这段代码实现了对 EEG 信号的**带
通滤波 (bandpass filter)**预处理，典型
用于去除低频漂移和高频噪声



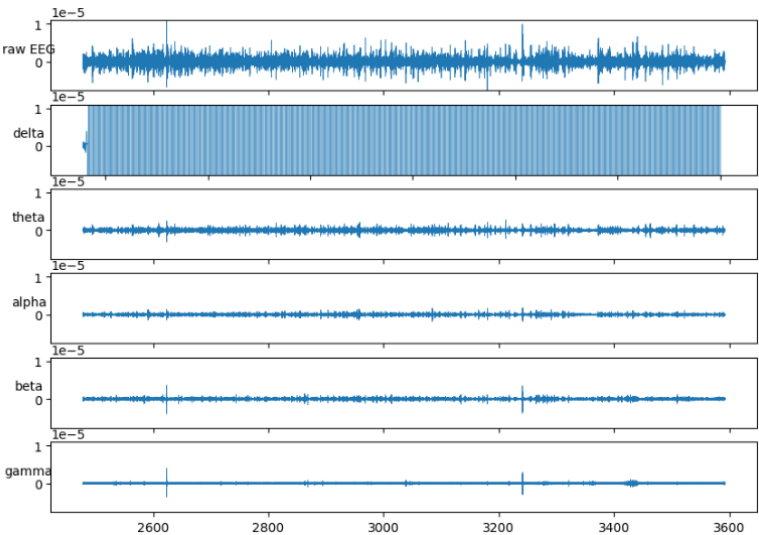
Этот код реализует преобразование Хилберта, позволяющее извлечь мгновенную амплитуду (амплитудную огибающую) ЭЭГ-сигнала. Такой метод широко используется в нейронауках и особенно подходит для анализа изменений огибающей сигнала, например ритмических мозговых волн.

Эта код реализует **Гильберта-преобразование (Hilbert transform)**,
используемое для извлечения **мгновенной амплитуды**
(**амплитудная огибающая**) сигнала. Этот метод
широко используется в нейронауках и особенно подходит для анализа
изменений огибающей сигнала, например ритмических мозговых волн.

Группа/波段	Диапазон частот (Гц)/频率范围 (Hz)	Физиологическое значение/生理含义
Delta	1–4	Глубокий сон/深度睡眠
Theta	4–8	Расслабление, легкий сон/放松、浅睡
Alpha	8–14	Состояние покоя/静息状态
Beta	14–30	Активность, бдительность/活动、警觉
Gamma	30–50	Познание высшего порядка/高阶认知

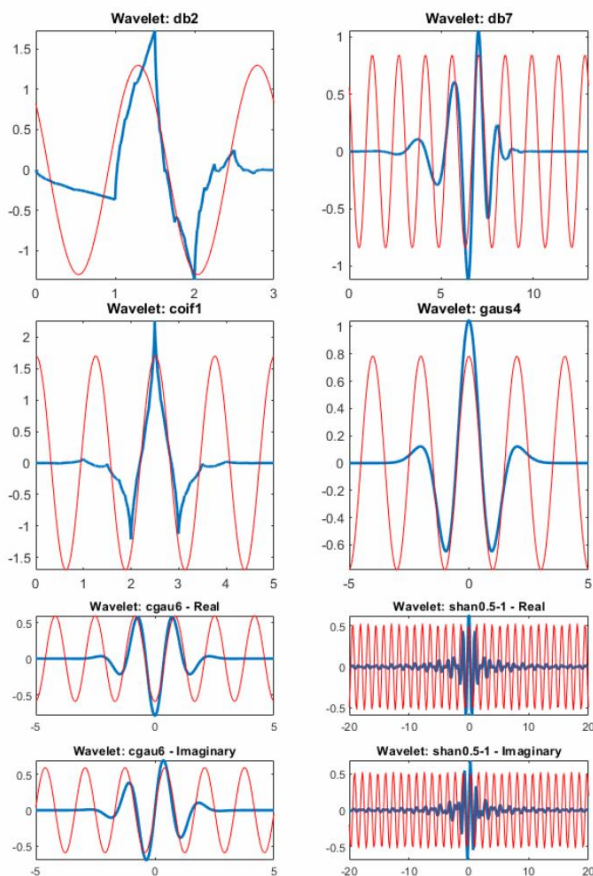
Это пять наиболее распространённых частотных диапазонов, применяемых в анализе ЭЭГ и соответствующих различным видам функциональной активности мозга.

这些是 EEG 分析中常用的 5 个频率区间，对应不同脑功能活



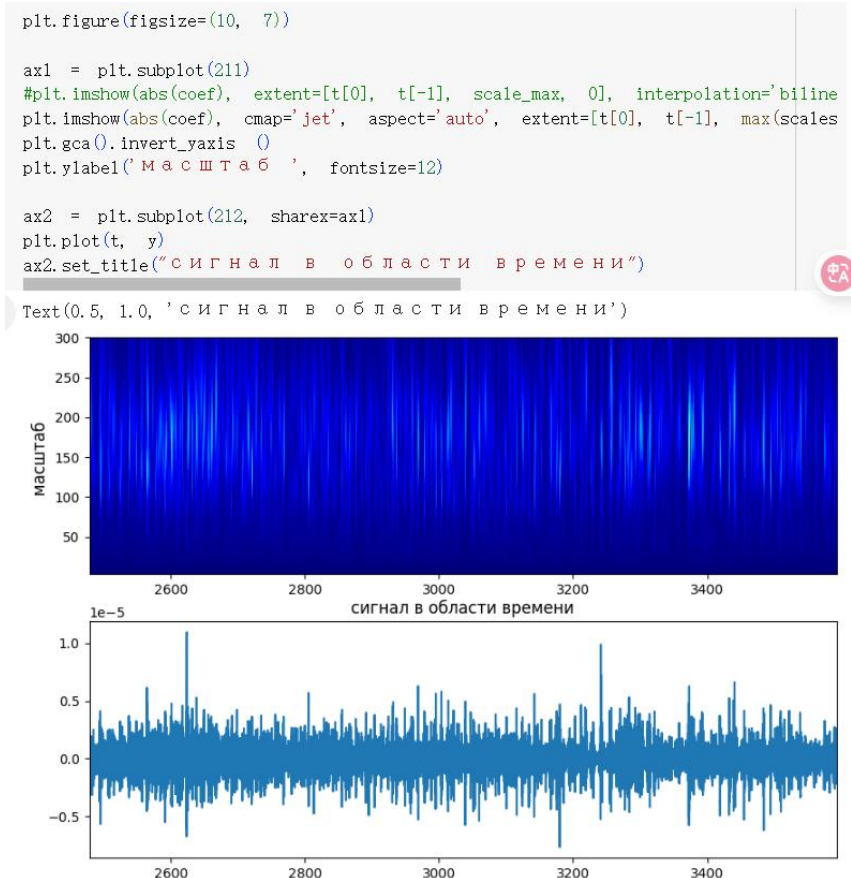
Цель данного кода совершенно очевидна: на одном графике (шесть строк подграфиков) изобразить исходный сигнал ЭЭГ и результаты фильтрации по пяти частотным диапазонам — δ , θ , α , β и γ . В анализе ЭЭГ это классический приём **визуализации полосового разложения (Band Decomposition Visualization)**.

Эта код的目标非常明确：将原始 EEG 信号及其 5 个频段（ δ 、 θ 、 α 、 β 、 γ ）的滤波结果绘制在一张图中（6 行子图），这在 EEG 分析中属于典型的**频带可视化（Band Decomposition Visualization）**方法。



Этот фрагмент кода выполняет непрерывное вейвлет-преобразование (Continuous Wavelet Transform, CWT) ЭЭГ-сигнала и задаёт диапазон масштабов, обеспечивающий нужное частотное разрешение. Метод подходит для времени-частотного анализа, позволяя отслеживать, как энергия сигналов различных частотных диапазонов распределяется по оси времени.

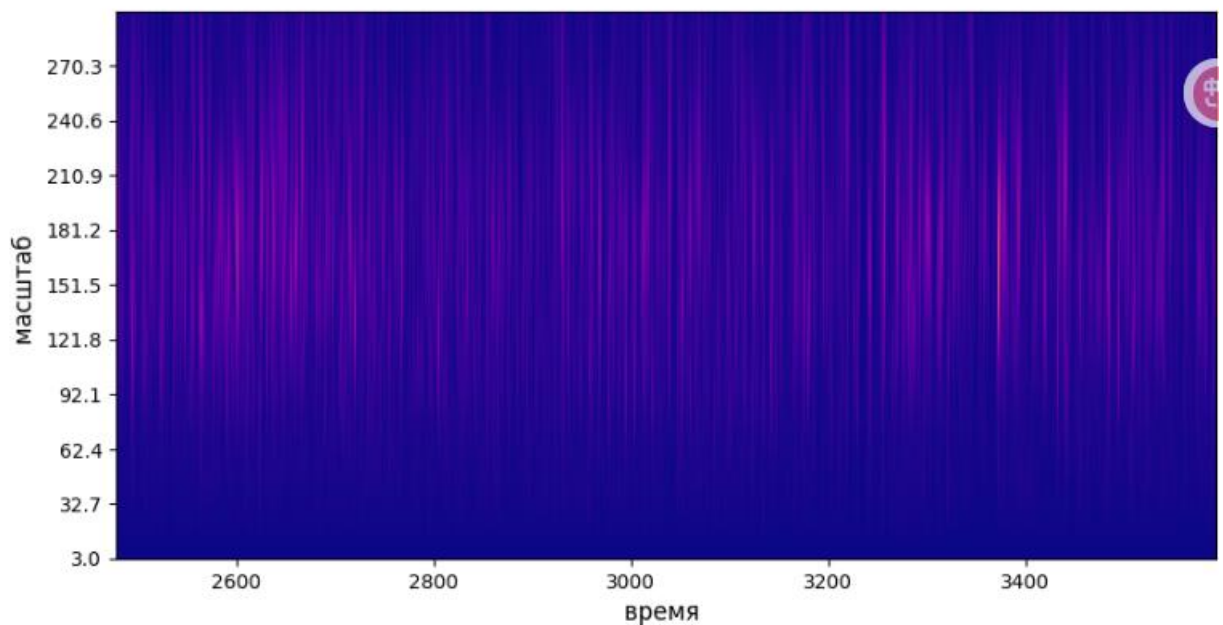
Эта код для EEG 信号进行连续小波变换 (Continuous Wavelet Transform, CWT)，并设置了变换的尺度范围以获得相应的频率分辨率。这种方法适用于时频分析，可以观察 EEG 中不同频段信号在时间轴上的能量分布。



Этот код строит очень полезное сочетание: временно-частотную карту непрерывного вейвлет-преобразования (CWT) и график соответствующего временного сигнала, что идеально подходит для детального анализа нестационарных сигналов, таких как ЭЭГ или ЭКГ.

Эта кодировка создала очень полезную непрерывную вейвлет-преобразование (CWT) временную частотную карту + соответствующий временной сигнал график, который подходит для ЭЭГ, ЭКГ и других временных сигналов для глубокого анализа.

```
#Plotting scalogram
plt.figure(figsize=(10, 5))
plt.imshow(abs(coef), extent=[t[0], t[-1], max(scales), min(scales)], interpolat
#plt.imshow(abs(coef), interpolation='bilinear', cmap='plasma', aspect='auto')
plt.gca().invert_yaxis ()
plt.yticks(np.arange(min(scales), max(scales), (max(scales) - min(scales))/10))
plt.ylabel('М а с ш т а б ', fontsize=12)
plt.xlabel('В р е м я ', fontsize=12)
plt.show()
```



На графике представлен скалограм (карта «время–масштаб») вейвлет-преобразования, также называемый вейвлет-тепловой картой; он показывает распределение энергии сигнала во времени и по различным масштабам (частотам).

Это важный метод визуализации при время-частотном анализе нестационарных сигналов, таких как ЭЭГ, ЭКГ и акустические данные.

Созданный является **маловолновым scalogram** (время-масштаб), также **маловолновым**, он показывает сигнал в разное время и в разном масштабе (частотах) энергии.

Это для ЭЭГ, ЭКГ, звука и др. нестационарных сигналов **время-частотный анализ** важный способ визуализации.

Структура папок изображений
классификации

- wavelet_img/Seizure/: 30/60/90 секунд до
приступа

wavelet_img/Healthy/: Случайные
сегменты без атак

Всего создано: 100 изображений (50
«Приступ» + 50 «Здоровье»)

классификации

- wavelet_img/Seizure/: 癫痫发作前
30/60/90 秒段

- wavelet_img/Healthy/: 随机无发作段

共生成: 100 张图像 (50 Seizure + 50
Healthy)

5) Построение модели CNN-LSTM

a) Структурное резюме:

5) CNN-LSTM 模型构建

a) 结构总结:

Conv2D(32) + BN + MaxPool

→ Conv2D(64) + MaxPool

→ Conv2D(128) + MaxPool

→ Flatten → Dense(64) + Reshape

→ LSTM(64) → Dense(32) → Dense(2)

б)Модуль CNN

Свёрточные нейронные сети в основном
используются для извлечения
пространственных локальных
особенностей на изображениях и
особенно подходят для распознавания
полосовой текстуры на вейвлет-
изображениях.

Многомасштабные частотно-временные
характеристики извлекаются с помощью
трех слоев свертки и, наконец, сводятся в

векторы с помощью Flatten и вводятся в
LSTM.

b) 卷积神经网络模块 (CNN)

卷积神经网络主要用于提取图像中的空
间局部特征, 尤其适用于小波图像中的
频带纹理模式识别。

通过三层卷积提取多尺度时频特征, 最
终通过 Flatten 展平成向量输入到
LSTM。

в) Сплюсчивание → Плотность (64):

- Преобразовать тензор пространственных признаков в одномерный вектор;
- Подключиться к слою Dense для предварительного нелинейного картирования;
- Добавлена регуляризация L2 для управления размером параметра:

$$L_{reg} = \lambda ||W||_2^2$$

с) Flatten → Dense(64):

- 将空间特征张量展平为一维向量;
- 接 Dense 层, 进行初步非线性映射;
- 添加 L2 正则化控制参数规模:

г) Изменение формы → LSTM(64):

- Поскольку LSTM требует последовательного ввода, измените плотный вывод на (1, 64);
- LSTM-единицы используются для моделирования «временных отношений» изображений (фактически последовательностей встроенных признаков).

d) Reshape → LSTM(64):

- 由于 LSTM 需要序列输入, 将 Dense 输出 reshape 为 (1, 64);
- LSTM 单元用于建模图像的“时间关系” (实际为嵌入特征序列)。

3. Обучение и оценка модели

1) Функция потерь и оптимизация

- Функция потерь: перекрестная
энтропия

3. 模型训练与评估

1) 损失函数与优化

- 损失函数：交叉熵

$$L = - \sum y_i \log(\hat{y}_i)$$

- Оптимизатор: Адам, скорость обучения 0.001
- Вес класса: сбалансировать соотношение эпилепсии и данных о состоянии здоровья
- EarlyStopping: отслеживание точности проверочного набора для предотвращения переобучения
- 优化器: Adam, 学习率 0.001
- Class weight: 平衡癫痫与健康数据比例
- EarlyStopping: 监控验证集准确率, 防止过拟合

Список литературы

[1] Гапанюк Ю. Е. Лабораторная работа «Разведочный анализ данных. Исследование и визуализация данных» [Электронный ресурс] // GitHub. — 2019. — Режим доступа: https://github.com/ugapanyuk/ml_course/wiki/LAB_EDA_VISUALIZATION (дата обращения: 13.02.2019)

[2] <https://www.kaggle.com/datasets>