

Второй отборочный этап

Индивидуальная часть

Задачи на программирование (Python)/ анализ данных

В задачах, которые вам предстоит решить, потребуется выполнить обработку и анализ различных сигналов. Рекомендуем использовать библиотеку matplotlib(<https://matplotlib.org/>) для визуализации предоставленных данных. Для знакомства с библиотекой вы можете ознакомиться с помощью данного учебника (<https://pythonworld.ru/novosti-mira-python/scientific-graphics-in-python.html>).

Для анализа сигналов вам потребуется использовать преобразование Фурье [1,2,3]. Библиотека numpy позволяет выполнять дискретное преобразование Фурье [1,2]. Использование моделей для классификации из библиотеки scikit-learn (<https://scikit-learn.org/stable/>) может позволить получить максимальный балл за 3 задачу.

Ввод исходных данных в 1 и 2 задачах осуществляется через stdin, проверка решения происходит с помощью stdout. Решение принимается в виде программы. В 3 задаче данные передаются в виде txt-файла. Ответ, соответствующий данному файлу, требуется ввести в течение 5 минут в виде строки.

Задача II.1.1.1. (5 баллов)

Сигнал $y(t)$ длительностью от 2 до 5 секунд представляет собой сумму двух сигналов: $f_1(t)$ и $f_2(t)$:

$$\begin{aligned}y(t) &= f_1(t) + f_2(t) \\f_1(t) &= A_1 \cdot \sin(2\pi \cdot \nu_1 \cdot t + \phi_1) \\f_2(t) &= A_2 \cdot \sin(2\pi \cdot \nu_2 \cdot t + \phi_2)\end{aligned}$$

Начальные фазы ϕ_1 и ϕ_2 заданы случайным образом, частота ν_1 удовлетворяет неравенству $2 \leq \nu_1 \leq 10$ и является натуральным числом, $\nu_2=1$, амплитуды удовлетворяют равенству $A_1 = 2 \cdot A_2$.

Сигнал оцифровывается с частотой 256 Гц. Результат оцифровки представлен последовательностью целых чисел лежащих в интервале от 0 до 255 и разделенных пробелом, амплитуда сигнала ≤ 50 . Напишите программу вычисляющую частоту ν_1 для поданного на вход сигнала.

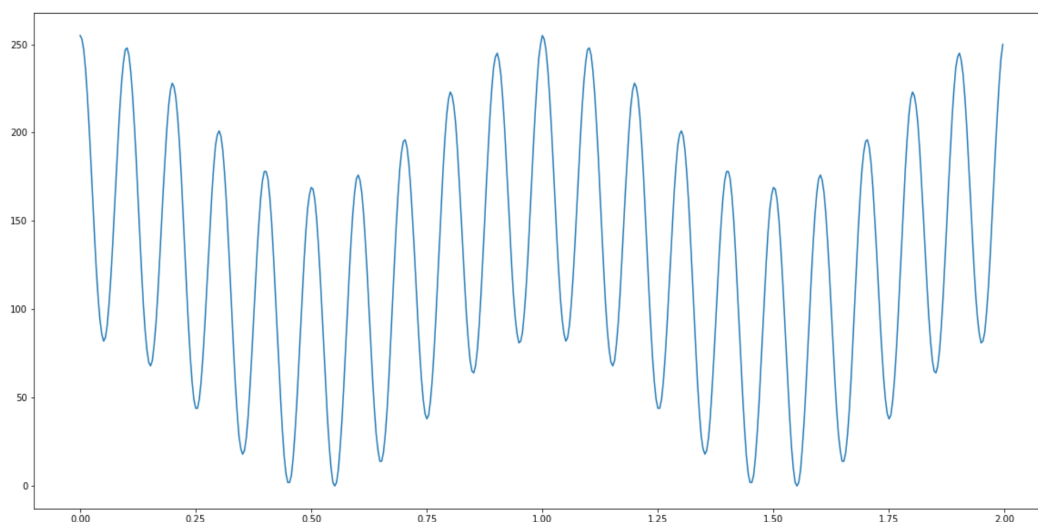


Рис. П.1.1: Пример сигнала с $\nu_1 = 10$ длительностью 2 с

Пример №1

Стандартный ввод

```

255 253 247 236 221 203 183 162 141 122 106 94 86 82 84 91 103 118 136
156 176 196 214 229 240 247 248 244 235 222 205 186 164 143 122 103 88
77 70 68 71 80 92 108 127 147 167 185 202 215 224 228 226 220 209 194
176 155 133 111 91 73 59 49 44 44 49 59 73 90 109 129 148 166 181 193
199 201 198 190 177 161 141 120 98 77 57 41 28 21 18 20 27 39 55 73 93
113 132 149 163 173 178 178 174 164 150 132 112 91 70 50 32 17 7 2 2 6
16 30 47 67 87 108 127 144 157 165 169 168 162 151 136 118 98 77 57 38
22 10 2 0 2 9 22 37 56 77 99 119 138 154 166 174 176 173 166 154 138
120 100 80 60 43 29 20 14 14 19 29 43 61 82 103 125 146 164 179 189 195
196 192 183 170 153 135 115 95 77 61 49 41 38 40 48 60 76 95 116 138
159 179 196 210 219 223 221 215 205 190 172 153 133 114 96 82 71 65 64
68 77 91 108 128 149 171 191 210 225 237 243 245 241 233 220 204 185
165 145 125 108 95 86 81 82 87 98 112 130 150 171 192 211 228 242 250
255 253 247 236 221 203 183 162 141 122 106 94 86 82 84 91 103 118 136
156 176 196 214 229 240 247 248 244 235 222 205 186 164 143 122 103 88
77 70 68 71 80 92 108 127 147 167 185 202 215 224 228 226 220 209 194
176 155 133 111 91 73 59 49 44 44 49 59 73 90 109 129 148 166 181 193
199 201 198 190 177 161 141 120 98 77 57 41 28 21 18 20 27 39 55 73 93
113 132 149 163 173 178 178 174 164 150 132 112 91 70 50 32 17 7 2 2 6
16 30 47 67 87 108 127 144 157 165 169 168 162 151 136 118 98 77 57 38
22 10 2 0 2 9 22 37 56 77 99 119 138 154 166 174 176 173 166 154 138 120
100 80 60 43 29 20 14 14 19 29 43 61 82 103 125 146 164 179 189 195 196
192 183 170 153 135 115 95 77 61 49 41 38 40 48 60 76 95 116 138 159 179
196 210 219 223 221 215 205 190 172 153 133 114 96 82 71 65 64 68 77 91
108 128 149 171 191 210 225 237 243 245 241 233 220 204 185 165 145 125
108 95 86 81 82 87 98 112 130 150 171 192 211 228 242 250

```

Стандартный вывод

10

Решение

```

1 import numpy as np
2 def get_spectrum(y):
3     """Функция возвращает для полученного в качестве аргумента сигнала два массива:
4     массив частот и массив амплитуд гармоник соответствующих данным частотам"""
5     Fs = 256.0 # частота оцифровки
6     n = len(y)
7     k = np.arange(n)
8     T = n/Fs
9     frq = k/T
10    frq = frq[range(n//2)]
11    Y = np.fft.fft(y)/n
12    Y = Y[range(n//2)]
13    return frq, abs(Y)
14 # считываем строку с тестовыми данными и преобразуем ее в массив целых чисел
15 data = list(map(int, input().strip().split()))
16 # получаем спектр сигнала
17 frq, ampl = get_spectrum(data)
18
19 # проходимся по массивам частот и амплитуд с конца, так как по условию неизвестная нам
20 ↪ частота первого сигнала
21 # больше известной нам частоты второго
22 for f, a in zip(frq[::-1], ampl[::-1]):
23     # порог амплитуды, по превышению которого судим о наличии данной компоненты в
24     ↪ сигнале, получили построив спектр
25     # сигнала из примера
26     if a > 1:
27         print(int(f))
28         break

```

Задача II.1.1.2. (10 баллов)

Электрмиограмма (ЭМГ) — это электрический сигнал, возникающий при сокращении мышечных волокон. ЭМГ регистрируется электродами с поверхности кожи, либо непосредственно с мышц. При кратковременном напряжении мышечных волокон на ЭМГ возникает характерная пучность.

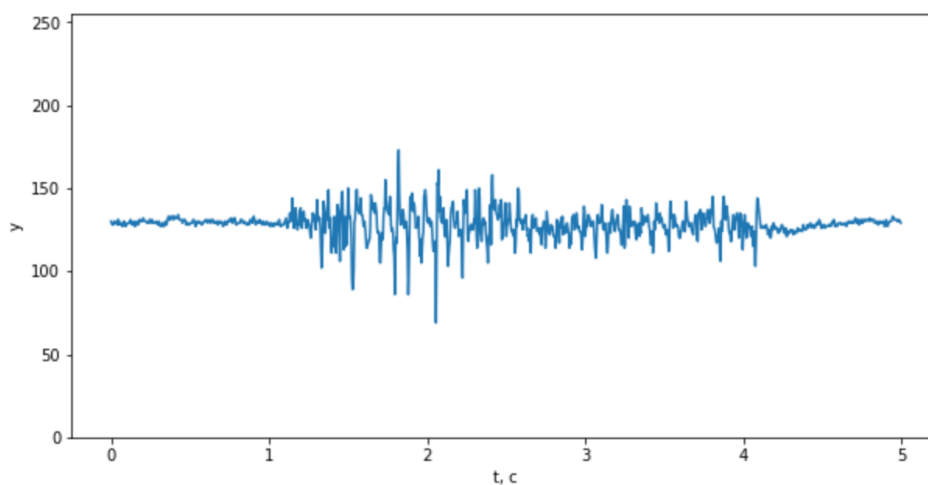


Рис. II.1.2: Сигнал ЭМГ

Пучность можно симитировать, переопределив значения на участке сигнала по следующей формуле: $y_i = 0.9 \cdot y_{i-1} + 0.1 \cdot z_i$, где z_i — i -й член выборки из равномерного распределения.

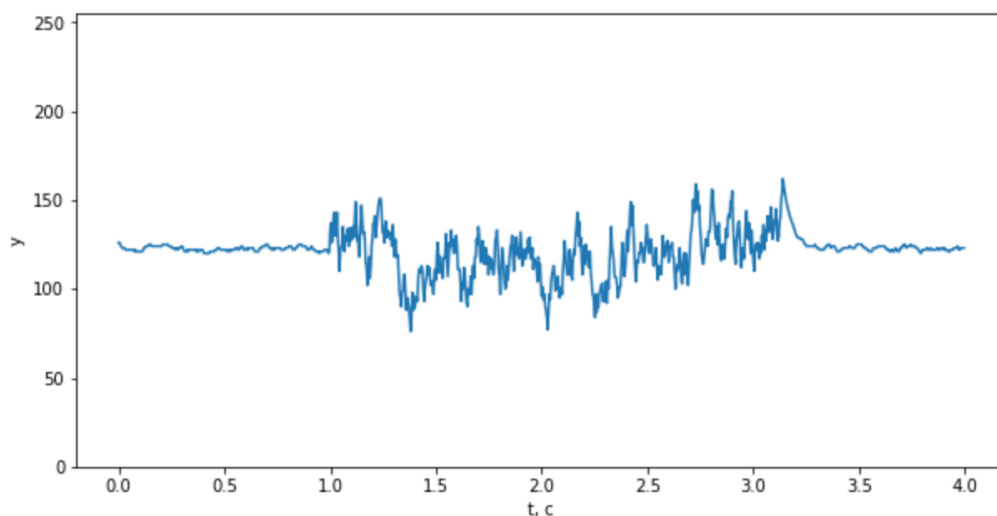


Рис. II.1.3: Равномерное распределение пропущенное через комплиментарный фильтр

Также значения на участке сигнала для имитации пучности можно определить как сумму двух гармоник: $y(t) = A_1 \cdot \sin(2\pi \cdot \nu_1 \cdot t + \phi_1) + A_2 \cdot \sin(2\pi \cdot \nu_2 \cdot t + \phi_2)$, где $\nu_1 \in [5, 9]$, $\nu_2 \in [10, 14]$.

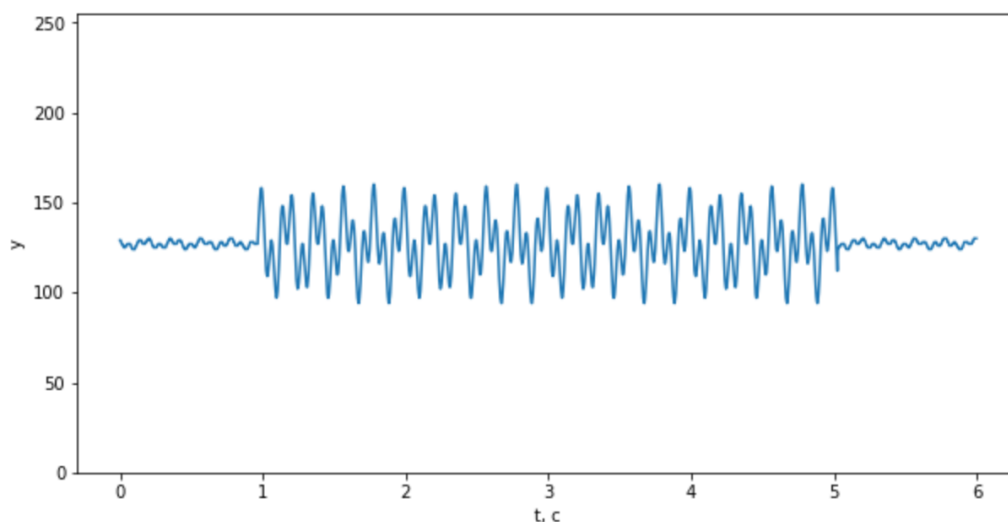


Рис. II.1.4: Сумма двух гармоник

Сигнал оцифровывается с частотой 250 Гц. Результат оцифровки представлен последовательностью целых чисел лежащих в интервале от 0 до 255 и разделенных пробелом. Длительность сигнала лежит в интервале от 4 до 6 секунд, сигнал содержит ровно одну пучность.

С образцами сигнала ЭМГ и его имитации вы можете ознакомиться по ссылке <https://drive.google.com/file/d/10Q31JS4VTA2X4TsiX5YiPBEvfJc4dzJc/view?>

`usp=sharing`. Напишите программу, определяющую, подается ли на вход настоящий сигнал ЭМГ или имитация. Для сигнала ЭМГ программа должна выводить в стандартный поток вывода 1, для имитации — 0.

Решение

Код для анализа образцов сигнала:

```

1  from pathlib import Path
2
3  for directory in Path('sample_2').iterdir():
4      plt.figure(figsize=(10,8))
5      plt.title(directory)
6
7      for file in directory.iterdir():
8          with open(file) as fin:
9              data = list(map(int, fin.readline().strip().split()))
10             frq, ampl = get_spectrum(data)
11             plt.plot(frq[1:], ampl[1:])
12 plt.show()
```

Строит три графика: график с реальными сигналами, график с сигналами из гармоник и график с равномерным распределением пропущенным через комплиментарный фильтр.

Из графиков находим, что сигнал сгенерированный как сумма двух гармоник является единственным типом сигнала амплитуда компонентов которого превышает значение 5. Эту особенность и используем для его определения.

Амплитуду гармоник сигнала сгенерированного с помощью равномерного распределения можно назвать убывающий с увеличением частоты. У реального сигнала ЭМГ интервал от 0 до 10 Гц не обладает наибольшей мощностью в спектре и в данном интервале у реального сигнала нет гармоник с амплитудой больше единицы.

Чтобы отличить один вид сигнала от другого, можно ввести и другие критерии, в т.ч. и относительные.

```

1  import numpy as np
2
3  def get_spectrum(y):
4      """Функция возвращает для полученного в качестве аргумента сигнала два массива:
5      массив частот и массив амплитуд гармоник соответствующих данным частотам"""
6      Fs = 256.0 # частота оцифровки
7      n = len(y)
8      k = np.arange(n)
9      T = n/Fs
10     frq = k/T
11     frq = frq[range(n//2)]
12     Y = np.fft.fft(y)/n
13     Y = Y[range(n//2)]
14     return frq, abs(Y)
15
16 # считываем строку с тестовыми данными и преобразуем ее в массив целых чисел
17 data = list(map(int, input().strip().split()))
18
19 # получаем спектр сигнала
```

```

20 frq, ampl = get_spectrum(data)
21
22 if max(ampl[1:]) > 5 or max(ampl[frq<10][1:]) > 1:
23     print(0)
24 else:
25     print(1)

```

Задача II.1.1.3. (15 баллов)

У большинства людей при закрытии глаз на электроэнцефалограмме (ЭЭГ) увеличивается амплитуда гармоник, лежащих в диапазоне от 8 до 12 Гц. В доступном по ссылке файле (https://drive.google.com/file/d/1GnhXZ1QEhb1bmk46mQh5keQsnoVQ_nF3/view?usp=sharing) находятся записи одноканальной ЭЭГ в следующем формате:

Type_0

Signal_0

Type_1

Signal_1

Signal_N — строка с сигналом ЭЭГ, представленным в виде последовательности целых чисел от -255 до 255 разделенных пробелом (частота оцифровки 256 Гц), Type_N — цифра 1, если Signal_N был записан с закрытыми глазами, 2 — если с открытыми.

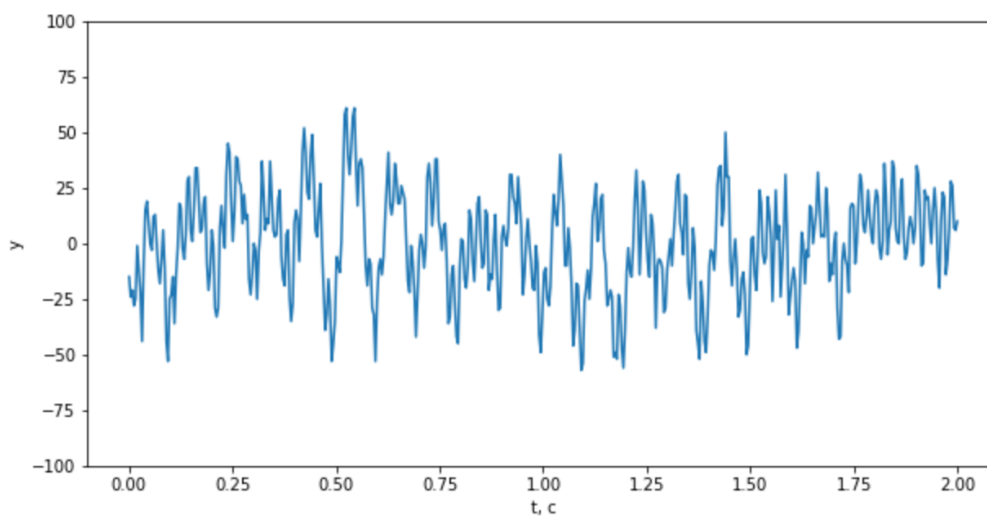


Рис. II.1.5: Образец сигнала для классификации

Разработайте модель для классификации типа сигнала ЭЭГ на основе предоставленного набора данных.

Для проверки качества модели вам будет предоставлен индивидуально сгенерированный набор данных в следующем формате:

TestSignal_0

TestSignal_1

Для TestSignal_N следует определить тип сигнала и предоставить ответ в виде

строки

Ans_0 Ans_1

где *Ans_N* — тип *TestSignal_N*.

После получения данных у вас будет 5 минут на отправку ответа. Оценка за задание выставляется путем умножения максимального балла за это задание на коэффициент *k*:

$$k = \max(0, \min(\frac{Accuracy - 0.55}{0.2}, 1))$$

где $Accuracy = \frac{N_{true}}{N_{total}}$ — отношение числа верно классифицированных образцов сигнала из тестовой выборки к общему числу образцов в тестовой выборке. Набор тестовых данных содержит 100 образцов сигнала.

Решение

```

1  import numpy as np
2  from sklearn.svm import SVC
3  from sklearn.model_selection import GridSearchCV
4
5
6  def get_spectrum(y):
7      """Функция возвращает для полученного в качестве аргумента сигнала два массива:
8      массив частот и массив амплитуд гармоник соответствующих данным частотам"""
9      Fs = 256.0 # частота оцифровки
10     n = len(y)
11     k = np.arange(n)
12     T = n/Fs
13     frq = k/T
14     frq = frq[range(n//2)]
15     Y = np.fft.fft(y)/n
16     Y = Y[range(n//2)]
17     return frq, abs(Y)
18
19
20 def get_features(signal):
21     """Функция возвращает вектор признаков для полученного сигнала.
22     Признаки - амплитуды гармоник соответствующих альфа-ритму"""
23     frq, ampl = get_spectrum(signal)
24     return ampl[(frq >= 8) & (frq <= 12)]
25
26 # Считываем обучающие данные, извлекаем признаки
27 with open('путь к файлу с данными для обучения', 'r') as file:
28     data = file.read().split('\n\n')
29     y_train, X_train = zip(*[d.split('\n') for d in data])
30     y_train = list(map(int, y_train))
31     X_train = [list(map(int, sample.split())) for sample in X_train]
32     X_train = [get_features(x) for x in X_train]
33
34 # Обучаем классификатор на доступных данных
35 parameters = {'C': [0.1, 0.5, 1, 3, 5, 10, 100]}
36 clf = GridSearchCV(SVC(), parameters, verbose=3, n_jobs=-1)
37 clf.fit(X_train, y_train)
38
39 # После получения удовлетворяющей нас точности на обучающей выборке, качаем тестовые
40 ↪ данные и
41 # предсказываем ответ
42 with open('путь к файлу с тестовыми данными') as file:

```

```

42     data = [np.array([int(j) for j in l.strip().split()]) for l in file.readlines() if
    ↪ l.strip()]
43 X_test = [get_features(d) for d in data]
44
45 # генерируем ответ
46 print(' '.join([str(i) for i in clf.predict(X_test)]))

```

Задачи для роли «Инженер-физиолог»

Задача II.1.2.1. (3 балла)

Как будет меняться картина на ЭЭГ, если синхронность работы двух рядом расположенных центров в головном мозге снизится (возрастет разность фаз для наиболее выраженной частоты для данных центров мозга)?

1. Будет наблюдаться увеличение амплитуды колебаний
2. Будет наблюдаться уменьшение частоты колебаний
3. Будет наблюдаться увеличение частоты колебаний
4. Будет наблюдаться уменьшение амплитуды колебаний

Решение

Правильный ответ — 4. (При увеличении разности фаз между различными источниками электрической активности в мозге на ЭЭГ будет наблюдаться уменьшение амплитуды колебаний).

Ответ: 4.

Задача II.1.2.2. (4 балла)

Какие стимулы и сдвиги в физиологических показателях будут вызывать чувство сытости после приема пищи?

1. Механическая стимуляция желудка
2. Понижение концентрации глюкозы в крови
3. Понижение концентрации холестерина в крови
4. Механическая стимуляция толстого кишечника
5. Механическая стимуляция тонкого кишечника
6. Повышение концентрации глюкозы в крови
7. Повышение концентрации холестерина в крови

Решение

Правильные ответы — 1, 6. (Ощущение сытости возникает при активации центра насыщения в гипоталамусе. Активация данного центра происходит при повышении уровня глюкозы в крови и растяжении стенок желудка.)

Ответ: 1; 6.

Задача II.1.2.3. (3 балла)

Что происходит с тонусом мышц плеча после приема пищи?

1. Тонус снижается
2. Тонус возрастает
3. Прием пищи не влияет на тонус мышц плеча

Решение

Правильный ответ — 1. (В процессе и после еды активируется парасимпатический отдел нервной системы, один из эффектов такой активации — снижение тонуса скелетной мускулатуры).

Ответ: 1.

Задача II.1.2.4. (3 балла)

Какой процесс начнется раньше других при задержке дыхания?

1. Снижение концентрации аргона в крови
2. Снижение концентрации кислорода в крови
3. Повышение концентрации углекислого газа в крови
4. Повышение концентрации азота в крови

Решение

Правильный ответ — 3. (Снижению концентрации кислорода препятствует депо кислорода в скелетной мускулатуре, поэтому при прекращении нормальной вентиляции легких накопление углекислого газа в крови будет предшествовать снижению уровня кислорода.)

Ответ: 3.

Задача II.1.2.5. (3 балла)

Через некоторое время после задержки дыхания у человека начинаются непроизвольные сокращения грудобрюшной диафрагмы (основной дыхательной мышцы). Большинство людей, если они не имеют специальной подготовки, которую проходят ныряльщики без акваланга, в этот момент, как правило, прекращают задержку дыхания. Изменение концентрации какого газа вызывает эти сокращения диафрагмы?

1. Азот
2. Аргон
3. Углекислый газ

4. Кислород
5. Угарный газ

Решение

Правильный ответ — 3. (Центр вдоха в продолговатом мозге активизируется при определенном уровне углекислого газа. Накопление в крови достаточного уровня углекислого газа будет стимулировать центр вдоха и последующее сокращение мышц вдоха.)

Ответ: 3.

Задача II.1.2.6. (5 баллов)

Зрение — один из основных источников информации об окружающем нас мире, глаз человека имеет некоторые особенности устройства, которые позволяют нам на основе дискретных изображений окружающего мира, отраженных на сетчатке получать полную картину того места, где мы находимся. Очень важна обработка информации, про которую мы будем вести речь в этом вопросе. Выберите структуры головного мозга, которые обрабатывают зрительную информацию с целью распознавания объектов.

1. Претектальная область ствола мозга
2. Зрительная кора
3. Латеральные коленчатые тела
4. Верхние холмики четверохолмия
5. Ядро зрительного тракта
6. Нижние холмики четверохолмия
7. Ядра вспомогательного зрительного тракта
8. Гипоталамус
9. Таламус

Решение

Зрительная информация обрабатывается в различных областях головного мозга. При этом существует функциональная дифференцировка между этими областями. В одних областях происходит распознавание объектов различной сложности, в других — детекция движения и т. д.

Ответ: 2; 3.

Задача II.1.2.7. (4 балла)

_____ ¹ обеспечивает предметное зрение, ее диаметр составляет от _____ ² до _____ ³ мм, что охватывает всего _____ ⁴ угловых градуса зрительного поля.

Решение

Предметное зрение обеспечивается областью с наибольшей плотностью фоторецепторов. Данная область имеет определенные нормальные размеры.

Ответ: 1 — Центральная ямка; 2 — 0,2; 3 — 0,4; 4 — 2.

Задача II.1.2.8. (5 баллов)

Внешний стимул, будь это запах готовой, вкусной еды, который вы почувствовали, или очень качественная фотография той самой еды, на которую вы смотрите, даже не чувствуя запаха может вызывать у человека два разных ответа: _____¹. В зависимости от типа ответа будут работать разные нейромедиаторы, в одном случае, если мы говорим про головной мозг, это обычно _____²

- глутамат;
- адреналин;
- серотонин;
- ацетилхолин;
- дофамин.

а в другом _____³,

- норадреналин;
- гамма-аминомасляная кислота;
- гистамин;
- аденозинтрифосфат;
- триптамин.

которые являются аминокислотами.

Решение

Стимул может вызывать типичную реакцию не только в ситуации реального присутствия, но и в ситуации воспоминания об этом стимуле или простого указания на него, например, при помощи фотографии. Ответ нервной системы будет различаться в зависимости от типа стимула. Нейромедиаторы для различного типа реакций также будут различаться.

Ответ: 1 — возбуждающий и тормозной; 2 — глутамат; 3 — гамма-аминомасляная кислота.

Командная часть

Командная задача. Часть 1

В командной задаче по физиологии вам предстоит ответить на три вопроса по условию, которое задано ниже на этой странице. На ответ по каждому вопросу

дается только одна попытка. Прочитайте условие внимательно, мы считаем, что все необходимые условия для решения этой задачи нами описаны и учтены.

Итак, перед Вами стоит задача оценить общее функциональное состояние пилота космического корабля, сохранность ресурсов его организма и уровень стресса при помощи минимально возможного количества аппаратуры и данных, поскольку связь с центром управления полетом плохая и передать большое количество данных невозможно.

Задача II.2.1.1. (5 баллов)

Укажите минимально необходимое и достаточное количество датчиков для интегральной оценки функционального состояния пилота

- 3
- 1
- 2
- 5
- 4

Решение

В общем случае для регистрации определенного электрофизиологического показателя нужно не менее двух датчиков: референтного и активного. В ряде случаев минимального количества датчиков может быть достаточно.

Ответ: 2.

Задача II.2.1.2. (5 баллов)

Укажите физиологический показатель/сигнал, который поможет решить поставленную в условии задачу наилучшим образом.

1. электрокардиограмма
2. электромиограмма
3. магнитоэнцефалограмма
4. кожно-гальваническая реакция
5. пульсоксиметрия
6. электроэнцефалограмма

Решение

В характеристиках ритмической активности сердца отражается как работоспособность самого сердца, так и статус вегетативной нервной системы, осуществляющей общую регуляцию деятельности внутренних органов. Поэтому характеристики ритмической активности сердца возможно использовать для интегральной оценки состояния систем регуляции внутренних органов.

Ответ: 1.

Задача II.2.1.3. Напишите ответ (5 баллов)

Укажите характеристику выбранного Вами физиологического сигнала/показателя, которая станет наилучшим интегральным индикатором общего функционального состояния пилота.

Решение

Ключевой характеристикой, отражающей интегральное состояние системы нервной регуляции внутренних органов, является характер variability сердечного ритма. Данный показатель называется индексом variability сердечного ритма или индексом Баевского.

Ответ: Индекс variability сердечного ритма.

Командная задача. Часть 2

В задачах данного блока вам предстоит классифицировать сигналы электроэнцефалограммы (ЭЭГ) и электромиограммы (ЭМГ). Помимо выбора алгоритма классификации и подбора его параметров, важную роль в достижении высокой точности на тестовых данных играет правильный выбор извлекаемых из сигнала признаков, поэтому участникам команды, отвечающим за программирование следует плотно взаимодействовать с участниками отвечающими за физиологию.

В предложенных далее задачах у вас будет 3 попытки на каждую задачу. Учитывается лучшая попытка. При расчете баллов за командную часть учитывается лучший результат участника команды за задачу.

Напоминаем, что вопросы по задачам вы можете задавать в чате второго этапа: https://t.me/joinchat/Bb0FDkb2e_BnwqtA4awLYw

Задача II.2.2.1. (5 баллов)

С предплечий нескольких участников эксперимента, повторявших определенный набор жестов, был записан сигнал ЭМГ. В доступном по ссылке (https://drive.google.com/file/d/1Nvk9tjL_t9PXT6TxXgHjgRqGnv3zwXfq/view?usp=sharing) файле находятся записи сигнала в следующем формате:

Class_0

Signal_00

Signal_01

Class_1

Signal_10

Signal_11

Signal_NM — строка с сигналом ЭМГ для M-того канала. Она представляет собой последовательность целых чисел от -300 до 300 разделенных пробелом (частота оцифровки 250 Гц), Class_N — номер жеста. Разработайте модель для классификации типа жестов по сигналу ЭМГ на основе доступного набора данных.

Для проверки качества модели вам для вас будет сгенерирован индивидуальный тестовый набор данных в следующем формате:

TestSignal_00

TestSignal_01

TestSignal_10

TestSignal_11

Для TestSignal_N0 TestSignal_N1 следует определить номер жеста и предоставить ответ в виде строки вида

Ans_0 Ans_1

Где Ans_N — тип номер N-того жеста в тестовых данных.

После получения данных у вас будет 5 минут на отправку ответа. Оценка за задание выставляется путем умножения максимального балла за это задание на коэффициент k:

$$k = \max(0, \min(\frac{Accuracy - 0.6}{0.2}, 1))$$

где $Accuracy = \frac{N_{true}}{N_{total}}$ - отношение числа верно классифицированных образцов сигнала из тестовой выборки к общему числу образцов в тестовой выборке. Набор тестовых данных, как и обучающих, содержит 87 образцов.

Примечание:

Сигнал намеренно зашумлен, эта особенность предоставленных данных не является ошибкой. Частота и амплитуда шума в тестовой и обучающей выборках совпадают.

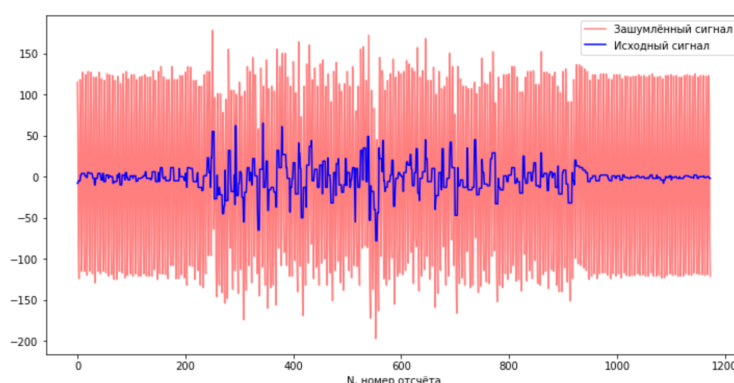


Рис. II.2.1: Сравнение исходного сигнала с сигналом предоставляемым участникам

Решение

```
1 import numpy as np
2 from scipy.signal import butter, lfilter
3 from sklearn.model_selection import GridSearchCV
```

```

4 from sklearn.svm import SVC
5
6 def filter_noise(signal):
7     """Функция для удаления шума из сигнала"""
8     b, a = butter(1, [48/125, 52/125], btype='bandstop')
9     return lfilter(b, a, signal)
10 def get_features(signal):
11     """Функция для извлечения признаков из сигнала"""
12     signal = filter_noise(signal)[100:] # Убираем начало сигнала, так как после
    ↪ фильтрации там присутствуют искажения.
13                                     # На пучность это не влияет.
14     f1 = np.sum(np.abs(signal))        # Извлекаем признаки из сигнала. С полным
    ↪ списком признаков для получения
15     f2 = f1 / len(signal)              # лучшей точности вы можете ознакомиться в
    ↪ http://repository.essex.ac.uk/14259/1/CES-513.pdf
16     f3 = np.sum(signal*signal)
17     f4 = f3 / len(signal)
18     f5 = np.sqrt(f4)
19     return np.array([f1, f2, f3, f4, f5])
20 X_train, y_train = [], []
21 # Считываем обучающие данные и извлекаем признаки из сигнала
22 with open('train_set.txt') as file:
23     fil = file.read()
24 for sample in fil.strip().split('\n\n'):
25     cls, ch0, ch1 = sample.split('\n')
26     y_train.append(int(cls))
27     ch0 = np.array([int(i) for i in ch0.split()], dtype=np.int)
28     ch1 = np.array([int(i) for i in ch1.split()], dtype=np.int)
29     x = np.concatenate([get_features(ch0), get_features(ch1)])
30     X_train.append(x)
31 # Обучаем классификатор, подбираем параметры.
32 parameters = {'C': [0.1, 0.5, 1, 3, 5, 10, 100]}
33 clf = GridSearchCV(SVC(), parameters, verbose=3, n_jobs=-1)
34 clf.fit(X_train, y_train)
35 X_test = []
36 # Считываем тестовые данные, извлекаем признаки.
37 with open('<файл с тестовыми данными>') as file:
38     x = file.read().split('\n\n')
39     for i in x:
40         ch0, ch1 = i.strip().split('\n')
41         ch0 = np.array([int(j) for j in ch0.split()], dtype=np.int)
42         ch1 = np.array([int(j) for j in ch1.split()], dtype=np.int)
43         x = np.concatenate([get_features(ch0), get_features(ch1)])
44         x = x.reshape(1, -1)
45         X_test.append(x)
46 X_test = np.concatenate(X_test)
47 # Получаем ответ
48 print(' '.join([str(k) for k in clf.predict(X_test)]))

```

Задача II.2.2.2. (15 баллов)

В данной задаче вам предстоит проанализировать 16-канальную запись электроэнцефалограммы, полученную в ходе эксперимента по детекции волны Р300. В ходе эксперимента испытуемый смотрел на дисплей с 36 объектами расположенными в виде матрицы 6х6. Изначально подсветка всех объектов была выключена.

Испытуемому подавалась команда ждать момента включения подсветки определенного объекта, после чего объекты начинали подсвечиваться группами по 6 штук

в случайном порядке (таким образом, за 6 циклов гарантированно загорался объект, подсветку которого ожидал участник эксперимента). Интервал между подсветкой различных групп был переменным.

Включение подсветки символа, за которым наблюдал испытуемый, вызывало возникновение волны P300 на ЭЭГ.

По ссылке(https://drive.google.com/file/d/1FBarUX1M-2Dml1-K8Y0dUBjuf_G3mexC/view?usp=sharing) вы можете скачать архив с данными записанными в ходе эксперимента. Сигнал разбит на отрезки длительностью 1 с, частота оцифровки — 512 Гц. Чтобы упростить детекцию P300 сигнал был пропущен через фильтр (<https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.firwin2.html#scipy.signal.firwin2>). Каждый отрезок сигнала находится в отдельном .csv файле, началу отрезка соответствует момент включения подсветки группы объектов. Столбцы в файле соответствуют отдельным каналам ЭЭГ: Fp1, Fp2, F3, AFz, F4, T7, Cz, T8, P7, P3, Pz, P4, P8, O1, Oz, O2.

Файлы организованы следующим образом: в директории target лежат отрезки сигнала, начало которых соответствует подсветке группы объектов включающей объект за которым наблюдаем испытуемый, в директории non-target — отрезки, начало которых соответствует подсветке группы без этого объекта, в директории test — файлы, для которых вам предстоит определить — относятся они к target или non-target.

target, non-target и test, относящиеся к одному испытуемому, лежат внутри одной директории, название которой соответствует номеру участника эксперимента.

Тестовые файлы имеют сквозную нумерацию. Ответ следует предоставить в виде строки из разделенным пробелом названий тестовых файлов (без расширения) относящихся к target. Например, если ваша модель определила что файлы 1.csv, 2.csv и 42.csv относятся к target, то ответ должен выглядеть как "1 2 42"(без кавычек). Оценка за задание выставляется путем умножения максимального балла на коэффициент k:

$$k = \min(\max(\frac{Acc - 0.5}{0.25}, 0), 1)$$

где $Acc = \frac{Sens+Spec}{2}$, Sens — отношение числа верно классифицированных target к общему числу target в тестовых данных, Spec — отношение числа верно классифицированных non-target к общему числу non-target в тестовых данных.

Эта задача оформлена на платформе Stepic как задача на данные, однако файл с данными понадобится исключительно для получения доступа к отправке ответа. Ответ рекомендуется вводить в поле на сайте, чтобы избежать возможных проблем с кодировкой. Чтобы избежать возможных сбоев также рекомендуется нажимать на "Нажмите, чтобы начать решать" уже после того как вы обработаете данные из архива и будете готовы ввести ответ.

Решение

```
1 from pathlib import Path
2 import numpy as np
3 from numpy import genfromtxt
4 from pyriemann.classification import MDM
5 from pyriemann.estimation import ERPCovariances
```



```

6 from sklearn.pipeline import make_pipeline
7 # Для каждого пользователя предсказываем классы в тестовых данных основываясь на
  ↳ обучающих данных только этого пользователя.
8 ans = set()
9
10 for user in range(5):
11     # Считываем данные для i-того пользователя
12     X_train, y_train = [], []
13     X_test, test_names = [], []
14
15     target = Path(f'dataset/{user}/target')
16     non_target = Path(f'dataset/{user}/non-target')
17     test = Path(f'dataset/{user}/test')
18
19     for file in target.iterdir():
20         data = genfromtxt(file, delimiter=',').T
21         X_train.append(data)
22         y_train.append(1)
23
24     for file in non_target.iterdir():
25         data = genfromtxt(file, delimiter=',').T
26         X_train.append(data)
27         y_train.append(0)
28
29     for file in test.iterdir():
30         data = genfromtxt(file, delimiter=',').T
31         X_test.append(data)
32         test_names.append(file.stem) # В массив test_names сохраняем названия файлов
  ↳ с тестовыми данными, т.о.
33
34                                     # j-тая запись из массива X_test была считана из
  ↳ "{test_names[j]}.txt"
35
36     # Обрезаем сигнал на 0.5 с основываясь на характерном времени регистрации P300
  ↳ после стимула
37     X_train = np.array(X_train)[: , : , :256]
38     y_train = np.array(y_train)
39     X_test = np.array(X_test)[: , : , :256]
40
41     # Для классификации классификацию по ближайшему центроиду (MDM из pyriemann). В
  ↳ качестве признаков используем
42     # ковариационные матрицы полученные pyriemann.estimation.ERPCovariances
43     # Данный пайплайн позволяет получить Acc = 0.84
44     clf = make_pipeline(ERPCovariances(estimator='lwf', classes=[1]), MDM())
45
46     # Обучаем модель и предсказываем классы для тестовых данных
47     clf.fit(X_train, y_train)
48     pred = clf.predict(X_test)
49     # Записываем в ans названия тех файлов, которые были отнесены к target
50     for p, n in zip(pred, test_names):
51         if p:
52             ans.add(n)
53     print(' '.join(ans))
54
55     # В тестовых данных к target относились следующие файлы:
56     # 1 6 9 13 15 32 44 45 46 49 64 70 74 76 81 89 94 98 105 113 118 129 138 143 144 147
  ↳ 154 158 164 166 175 186 190 191
57     # 193 195 203 204 208 211 216 220 223 228 237 246 248 264 282 294 298 299 302 307 308
  ↳ 316 318 320 338 354 361 372 382
58     # 385 386 387 389 392 393 403 410 415 416 417 425 448 449 450 457 458 467 480 484 494
  ↳ 496 520 522 531 532 535 539 541
59     # 559 561 568 570 576 579 581 583 590 603 607 619 623 '

```