

Реализация аппаратного модуля линейного дискриминанта Фишера для классификации биомедицинских сигналов

Д. А. Прилуцкий¹, А. П. Гусев²

МГТУ им. Н.Э. Баумана

¹denis.prilutsky@outlook.com, ²xitren@ya.ru

Д. А. Баловнев

Институт электронных управляющих машин

им. И.С. Брука

balovnev_d@ineum.ru

Аннотация. Приведен подход к реализации аппаратного модуля классификатора биомедицинских сигналов на основе линейного дискриминанта Фишера. Дана оценка сложности алгоритма обучения классификатора. Разработана структура построения классификатора на основе микросхем FPGA, CPLD, позволяющая распараллеливать потоки данных на заданное количество отведений.

Ключевые слова: анализ биомедицинских данных; линейный дискриминант Фишера; Pipeline; FPGA; CPLD

I. ВВЕДЕНИЕ

Классификация многомерных данных широко применяется в биологических и медицинских исследованиях [1, 2, 3]. Одним из эффективных алгоритмов для разделения сигналов на два класса является линейный дискриминант Фишера (ЛДФ), который уменьшает размерность признакового пространства с исходного до одного измерения путем проектирования многомерных данных на прямую. Качество обучения классификатора зависит от количества признаков. Увеличение количества признаков, как будет показано далее, ведет к возрастанию вычислительной сложности алгоритма, поэтому, для уменьшения размерности, может применяться предобработка входных данных, например, децимация входного сигнала. Но такой подход может быть неприемлем в системах, где полезный сигнал может иметь амплитуду, сравнимую с амплитудой шумов. На рис. 1 показан спектр сигнала ЭЭГ (электроэнцефалографа), полученный в результате эксперимента.

В медицинских измерениях с использованием носимых устройств, основанных на микроконтроллерных системах, реализация классификатора практически не применима, а в устройствах, где количество отведений может быть более 128, – вовсе становится невозможным из-за ограниченной производительности таких систем.

Для разрешения данной проблемы возможно использование микросхем FPGA (Field-Programmable Gate Array), CPLD (Complex Programmable Logic Device), позволяющих проектировать параллельную архитектуру вычислений для многомерных данных.

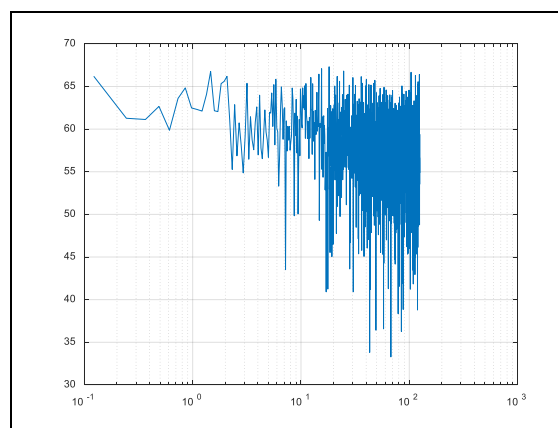


Рис. 1. Спектр сигнала ЭЭГ, полученный в результате эксперимента

Целью данной работы является разработка аппаратного модуля, который реализует классификацию, основанную на ЛДФ, для одного отведения (измерителя) входного сигнала.

Для проверки целесообразности построения такого классификатора, рассмотрим алгоритм обучения классификатора на основе ЛДФ в эксперименте с применением метода Р300 [4] и дадим оценку вычислительной сложности алгоритма.

II. ОПИСАНИЕ ЭКСПЕРИМЕНТА

Метод Р300 имеет два режима работы: режим обучения и режим управления.

В режиме обучения на мониторе компьютера или на специальном устройстве предъявителя выдаются стимулы – изображения, которые должны вызвать в затылочных зонах мозга реакцию Р300 – вызванный потенциал (ВП), появляющийся в среднем после 300 мс [4] и имеющий положительную амплитуду на ЭЭГ. Р300 – это реакция на ожидаемый визуальный стимул, предъявляющийся в определенное время в определенной позиции на экране.

На рис. 2 показаны изображения, показываемые человеку в режиме обучения и управления.

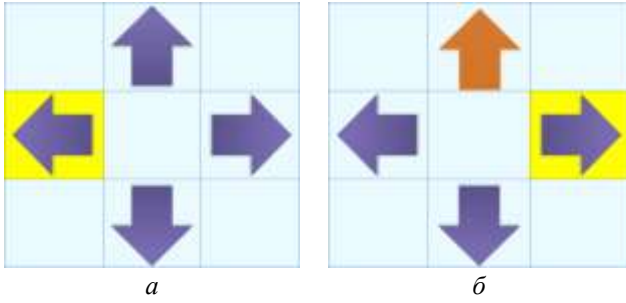


Рис. 2. Изображения, предъявляемые человеку в режиме обучения и управления: а) фоновая подсветка картинки – стимул; б) коричневым выделяется целевое изображение

Предъявляемым стимулом в нашем эксперименте является случайно изменяющийся фон на изображениях стрелок.

Коричневым цветом подсвечивается стрелка, которая в данный момент времени является целевой – на которую человеку нужно смотреть в течение всего периода обучения. На стимул в этой позиции возникает реакция Р300. На каждом целевом изображении количество предъявляемых стимулов (число загораний фона) равно ~5–7. Выделение целевого изображения и предъявление стимула производится независимо и в случайном порядке.

Запись ЭЭГ идет непрерывно. Синхронно с ней записываются данные стимулов – время и позиция предъявления.

В режиме обучения необходимо зарегистрировать и разделить при помощи классификатора два типа реакций: Р300 присутствует; Р300 отсутствует. Время обучения длится в течение 5–7 минут. Следующим этапом идет режим управления.

В режиме управления производится различение на присутствие в сигнале реакции Р300 или ее отсутствия с помощью обученного на предыдущем этапе классификатора. Важным моментом здесь является то, что мы все также продолжаем предъявлять те же изображения и стимулы и записывать временные метки их предъявления. Человек сосредотачивает свое внимание на выбранной картинке – детектируется реакция Р300 и, в случае если количество реакций более 3–5, подсвечивается выбранная стрелка.

Далее, опишем подробнее подготовку данных измерений для обучения классификатора на основе ЛДФ.

III. ПОДГОТОВКА ДАННЫХ ИЗМЕРЕНИЙ

Мы разбиваем ЭЭГ на фрагменты относительно времени появления стимулов на предъявителе.

Зададим входной вектор признаков для последующего анализа:

$$S = [\bar{u}_0 \quad \bar{u}_1 \quad \dots \quad \bar{u}_{m-1}]^T,$$

где $\bar{u}_i = (u_0 \quad u_1 \quad \dots \quad u_{n-1})$ – измеренные напряжения с поверхности кожи головы человека;

m – размерность вектора признаков;

n – количество измерений. Данное число в эксперименте $n = 125$.

Для обучения классификатора требуется разделить вектора с целевыми ВП и нецелевыми. Для разделения данных используется вектор, содержащий информацию о том, в какие моменты срабатывал предъявитель визуальных сигналов, благодаря этому возможно выделение векторов с ВП. Получим два класса – две матрицы, содержащие последовательности измерений с целевыми стимулами, S_1 , и нецелевыми, S_2 :

$$S_1 = [\bar{u}_{p_0} \quad \bar{u}_{p_1} \quad \dots \quad \bar{u}_{p_{m_1-1}}]^T;$$

$$S_2 = [\bar{u}_{d_0} \quad \bar{u}_{d_1} \quad \dots \quad \bar{u}_{d_{m_2-1}}]^T,$$

где $p_0, p_1, \dots, p_{m_1-1}$ – индексы признаков исходной матрицы, содержащих целевые стимулы;

m_1 – количество признаков, содержащих целевые стимулы; размерность полученного вектора;

$d_0, d_1, \dots, d_{m_2-1}$ – индексы признаков исходной матрицы, содержащей нецелевые стимулы;

m_2 – количество последовательностей, не содержащих целевые стимулы; размерность полученного вектора.

Опишем процесс обучения классификатора.

IV. ОБУЧЕНИЕ КЛАССИФИКАТОРА

Рассчитаем средние значения, \bar{S}_1 и \bar{S}_2 , для каждого класса:

$$\begin{aligned} \bar{S}_1 &= [\bar{U}_{p_0} \quad \bar{U}_{p_1} \quad \dots \quad \bar{U}_{p_{m_1-1}}]^T = \\ &= \frac{1}{m_1} \left[\sum_{i=0}^{n-1} u_{p_0,i} \quad \sum_{i=0}^{n-1} u_{p_1,i} \quad \dots \quad \sum_{i=0}^{n-1} u_{p_{m_1-1},i} \right]^T; \\ \bar{S}_2 &= [\bar{U}_{d_0} \quad \bar{U}_{d_1} \quad \dots \quad \bar{U}_{d_{m_2-1}}]^T = \\ &= \frac{1}{m_2} \left[\sum_{i=0}^{n-1} u_{d_0,i} \quad \sum_{i=0}^{n-1} u_{d_1,i} \quad \dots \quad \sum_{i=0}^{n-1} u_{d_{m_2-1},i} \right]^T. \end{aligned}$$

Далее, рассчитаем ковариационные матрицы для каждого класса:

$$\Sigma_1 = \text{cov}(S_1); \Sigma_2 = \text{cov}(S_2).$$

Полученные ковариационные матрицы имеют размерность m_1 и m_2 соответственно.

Рассчитаем внутриклассовую матрицу разброса:

$$S_w = \Sigma_1 + \Sigma_2.$$

Полученная матрица имеет размерность $m_1 = m_2$.

Вычисляем вектор W :

$$W = S_w^* (\bar{S}_1 - \bar{S}_2),$$

где $S_w^* = S_w^T \times (S_w \times S_w^T)^{-1}$ – псевдоинверсная матрица, размерность которой равна размерности матрицы внутриклассового разброса.

Находим проекции двух классов на вектор W :

$$Y_1 = W^T \times S_1;$$

$$Y_2 = W^T \times S_2.$$

Размерность полученных векторов равна m_1 и m_2 соответственно. Определим порог для двух классов. Найдем средние значения для первого и второго класса:

$$M_1 = \frac{1}{m_1} \sum_{i=0}^{m_1-1} y_{p,i}; M_2 = \frac{1}{m_2} \sum_{i=0}^{m_2-1} y_{d,i},$$

получим значение порога для нахождения принадлежности вектора к классу:

$$w_N = \frac{|M_2 - M_1|}{2}.$$

Далее, для классификации произвольного вектора X производится его перемножение на вектор W и сложение с полученным порогом:

$$d(X) = \sum_{i=0}^{N-1} w_i \times x_i + w_N.$$

Решение о принадлежности к классу производится по формуле:

$$\text{sgn}[d(X)] = \begin{cases} \Omega_1; \\ \Omega_0; \\ \Omega_2, \end{cases}$$

где Ω_1, Ω_2 – первый и второй класс;

Ω_0 – вектор не относится ни к первому, ни ко второму классу.

Далее, произведем оценку сложности применяемых вычислений.

V. ВЫЧИСЛИТЕЛЬНАЯ СЛОЖНОСТЬ АЛГОРИТМА

Приведем количество арифметических операций в нотации «О».

Перемножение матриц:

$$O_{mult}(m, n, p) = m \times n \times p,$$

где m – количество строк первой матрицы;

n – количество столбцов первой матрицы и количество строк второй;

p – количество столбцов второй матрицы.

Сложение, вычитание матриц: $O_{sum}(m, n) = m \times n$.

Вычисление обратной или псевдоинверсной матрицы:

$$O_{pinv}(m) = m^3.$$

Вектор средних значений матрицы:

$$O_{Expected}(m, n) = m \times n.$$

Вычисление ковариационной матрицы:

$$O_{cov}(m) = m^3 + m.$$

Комбинируя данные зависимости для вычисления вектора W получаем суммарное количество операций. Зависимость количества арифметических операций от размерности входных данных показана на рис. 3.

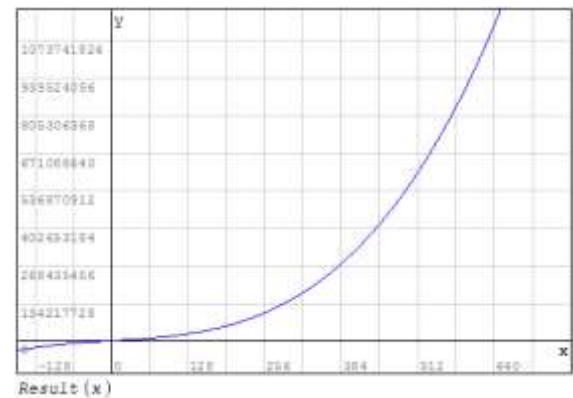


Рис. 3. Зависимость количества операций от размерности входного вектора признаков

Как видно из рис. 3, количество операций нелинейно зависит от размерности вектора признаков. Для увеличения производительности классификатора требуется разработать структуру модуля, на основе существующих архитектур.

VI. ВЫБОР АРХИТЕКТУРЫ СИСТЕМЫ

Для использования преимуществ CPLD была выбрана архитектура Pipeline, являющейся простой и надежной конвейерной архитектурой. Она может состоять из любого количества компонентов (Filter), которые преобразуют или фильтруют данные, прежде чем передавать их через каналы (Pipe) на другие компоненты. Компоненты такой системы начинают свою работу по приходу всех данных с канала. Архитектура часто используется как простая последовательность, но ее также можно использовать для более сложных структур. На рис. 4 показан пример построения системы на основе данной архитектуры.

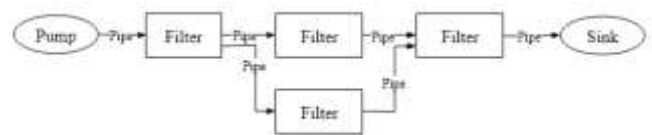


Рис. 4. Пример системы на основе архитектуры Pipeline

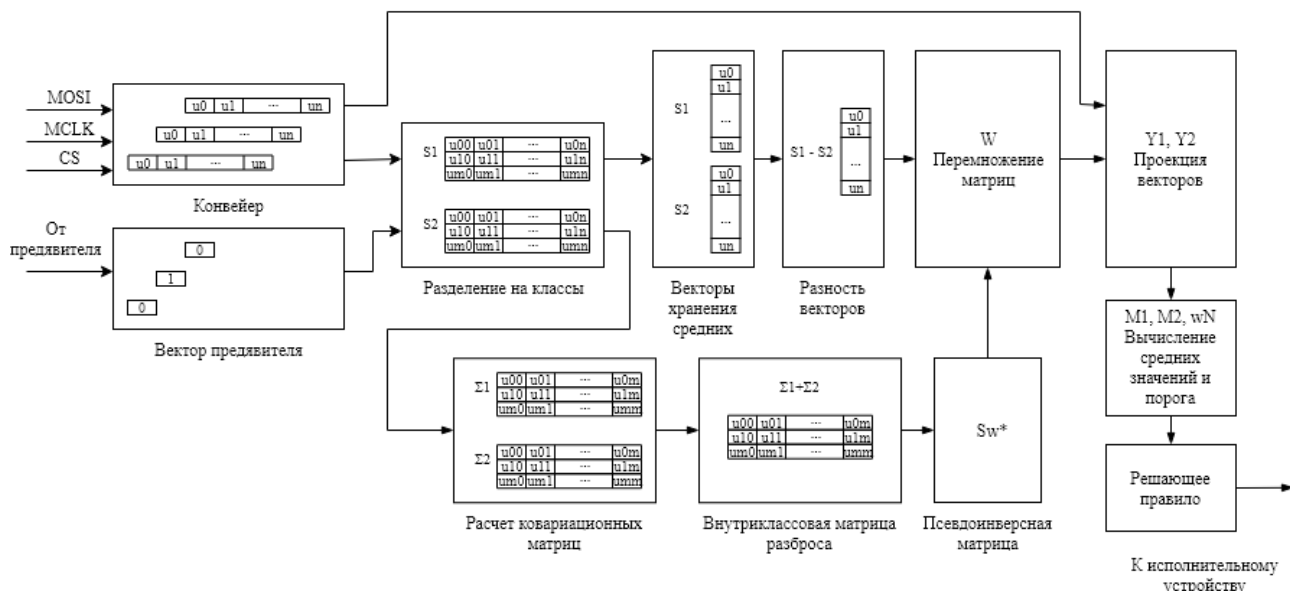


Рис. 5. Структура модуля классификатора на основе ЛДФ

Pump является источником данных. Это может быть статический текстовый файл или устройство ввода, постоянно генерирующее новые данные. Sink или потребитель является приемником данных. Это может быть другой файл, база данных или экран компьютера. В нашем случае источником данных является интерфейс SPI. Приемник – последующий каскад фильтра, либо вывод данных на микроконтроллер.

Далее, опишем предлагаемое решение.

VII. ПРЕДЛАГАЕМОЕ РЕШЕНИЕ

На рис. 5 показана предлагаемая структура модуля классификатора, реализуемая в CPLD. Данная структура может быть задана с использованием языка описания аппаратуры интегральных схем, такого как VHDL (Very High Speed Integrated Circuits Hardware Description Language), либо Verilog HDL.

Конвейер – накапливает данные, приходящие с порта SPI (Serial Peripheral Interface). Далее, происходит разделение векторов с использованием меток предъявителя, которые принимаются синхронно с приходом данных конвейера. Хранение векторов производится в синтезированной памяти микросхемы. После разделения векторов на классы, производится пересчет псевдоинверсной матрицы исходя из свойства ковариационной матрицы:

$$\text{cov}(X, Y) = E(X \times Y) - \bar{X} \times \bar{Y}.$$

и вектора, хранящих средние значения. В результате перемножения матриц получим матрицу коэффициентов W. С приходом специального сигнала статуса окончания обучения, начинается проецирование входных векторов на полученную прямую W и по решающему правилу производится классификация: информация

передается по последовательному порту микроконтроллеру.

VIII. ПОЛУЧЕННЫЕ РЕЗУЛЬТАТЫ

При синтезировании конфигурации для микросхем фирмы Lattice Semiconductor серии LCMX02 количество LUTs (look-up table) для формирования логики модуля составило <1000 для одного блока классификатора. Хранение промежуточных вычислений в синтезированной памяти микросхемы составило 33 EBR (Embedded Block RAM).

Таким образом, был получен аппаратный модуль классификатора на основе ЛДФ.

СПИСОК ЛИТЕРАТУРЫ

- [1] Aboul Ella Hassanien, Ahmad Tahar Azar, Brain-Computer Interfaces. Intelligent Systems Reference Library Vol.74, 416 p.
- [2] Методы статистической обработки медицинских данных: Методические рекомендации для ординаторов и аспирантов медицинских учебных заведений, научных работников / сост.: А.Г. Кочетов, О.В. Лянг., В.П. Масенко, И.В. Жиров, С.Н. Наконечников, С.Н. Терещенко. М.: РКНПК, 2012. 42 с.
- [3] Guang-Zhong Yang, Body Sensor Networks, 2014.
- [4] Juri Kropotov, Quantitative EEG, Event-Related Potentials and Neurotherapy, 2008.
- [5] Rhythms of the Brain / G. Buzsaki, - Oxford: Oxford University Press, 2006.
- [6] Pattern classification and scene analysis / Richard O. Duda, Peter E. Hart, 2001.
- [7] M. Karkooti, J. R. Cavallaro and C. Dick, FPGA Implementation of Matrix Inversion Using QRD-RLS Algorithm, 2005.
- [8] A. Irturk, S. Mirzaei, and R. Kastner, An efficient FPGA implementation of scalable matrix inversion core using QR decomposition, Univ. California, San Diego, San Diego, CA, USA, UCSD Tech. Rep. CS2009-0938, 2009.