Задача распознавания жестов

# Мотивация

Наша команда стремится, чтобы 100% людей без рук ежедневно пользовались доступными функциональными протезами рук с высоким уровнем сервиса, делая свою жизнь удобнее, ярче и счастливее.

Компания «Моторика» исследует и разрабатывает технологии на стыке медицины и робототехники с 2014 года. На данный момент мы выпускаем два типа протезов: активные тяговые протезы Киби и бионические протезы Инди и Манифесто. Протезы подходят для детей и взрослых с ампутациями верхних конечностей на уровне кисти и предплечья.

Также компания решает различные R&D задачи, в том числе которые находятся на стыке машинного обучения и человеко-машинных интерфейсов. На пересечении данных областей могут быть найдены или открыты совершенно новые подходы к управлению протезами.

В ходе будущей стажировки компания Моторика совместно со Skillfactory предлагает участникам попробовать свои силы в решении задач машинного обучения в области протезирования.

# Сбор данных

## Сетап

Назовем Пилотом человека, с которого производится сбор данных.

На руку пилота надета оптомиографическая манжета (дальше будем именовать ее как ОМГ-манжета), главным элементом которой являются оптомиографические датчики.

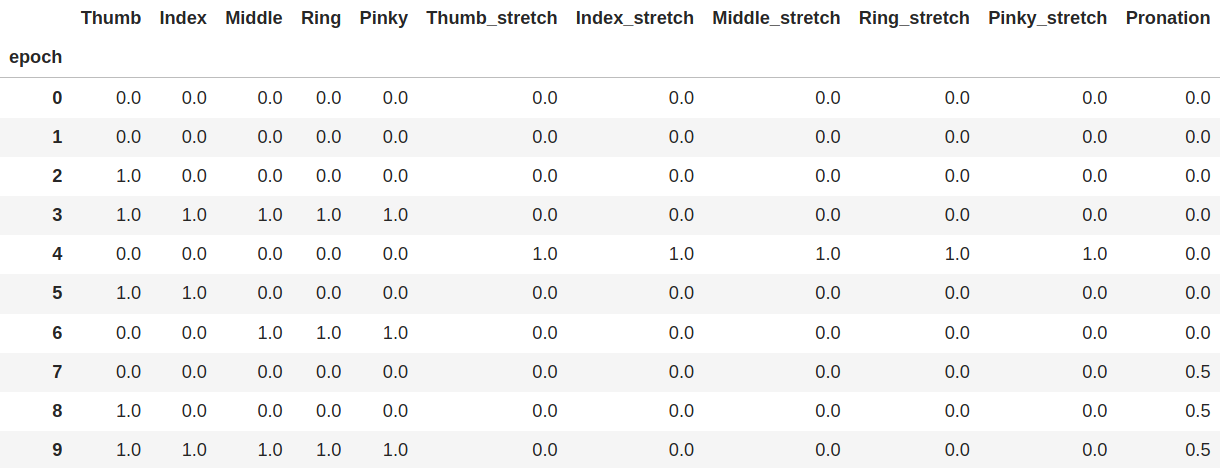


*Изображение* — (a) ОМГ-манжета вид с внутренней стороны, где располагаются датчики; (b) ОМГ-манжета установленная на руку пилота перед процедурой сбора данных; (с) место контакта датчиков ОМГ-манжеты с рукой пилота.

В ходе процедуры сбора данных ОМГ-манжета с определенной частотой фиксирует, полученные от ОМГ-датчиков значения.

## Протокол сбора данных

Чтобы унифицировать процедуру сбора данных собирается некоторый *Протокол*. Протокол - это последовательность жестов, которые выполняет пилот во время процедуры сбора данных.



*Изображение* — табличное представление протокола (первые 10 строк).

Столбец ‘epoch’ обозначает номер выполняющегося жеста, может быть использован для дальнейшего сопоставления с ОМГ-данными (по столбцу ‘SYNC’).

Каждая строчка протокола содержит в себе команды, которую должен выполнить пилот. Первые 5 столбцов [Thumb, …, Pinky] обозначают сгибаются ли соответствующие пальцы, а вторые 5 столбцов [Thumb\_stretch, …, Pinky\_stretch] обозначают вытягиваются ли соответствующие пальцы;

Разберем жесты закодированные в протоколе:

* Neutral/NOGO (все нули) - нейтральное положение кисти, кисть в расслабленном состоянии;
* Thumb (Thumb равен 1) - сгиб большого пальца;
* Grab (Thumb, Index, Middle, Ring, Pinky равны 1) - схват, кисть сжата в кулак;
* Open (Thumb\_stretch, Index\_stretch, Middle\_stretch, Ring\_stretch, Pinky\_stretch равны 1) - открытая ладонь, пальцы выпрямлены;
* OK (Thumb, Index равны 1) - жест “Окей”;
* Pistol (Middle, Ring, Pinky равны 1) - жест “Пистолет” ;

Протокол цикличен, есть некоторая подпоследовательность жестов, которая повторяется.

На текущем шаге у нас есть необходимое оборудование для сбора данных, заданная последовательность жестов к выполнению пилотом. Теперь можно собирать данные. Но чтобы облегчить задачу пилота, можно использовать протокол для сбора данных в парадигме Motion Match.

## Motion Match

Относительно специфики задачи motion match используется следующим образом:

* кто-то / что-то отдает команду пилоту на выполнение жеста;
* пилот выполняет движение, согласно отданной команде.

В нашем конкретном случае используется виртуальная рука, которая выполняет жесты согласно заданному ранее протоколу. Пилот повторяет жесты за виртуальной рукой с некоторой естественной задержкой. Такой подход сбора данных позволяет пилоту больше сконцентрироваться на воспроизведении жестов.



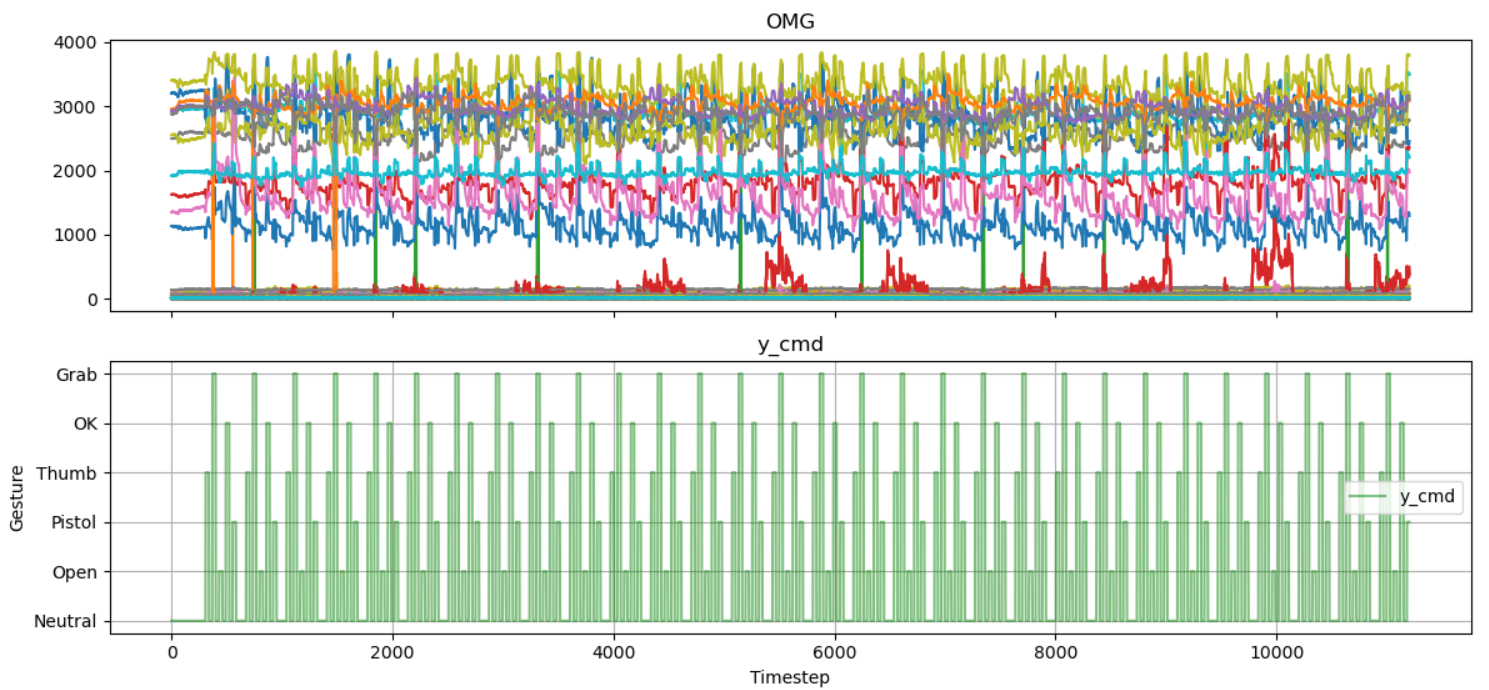
*Видео* — пример работы виртуальной руки согласно заданному протоколу

В текущей парадигме переход между жестами выполняется через нейтральное состояние:

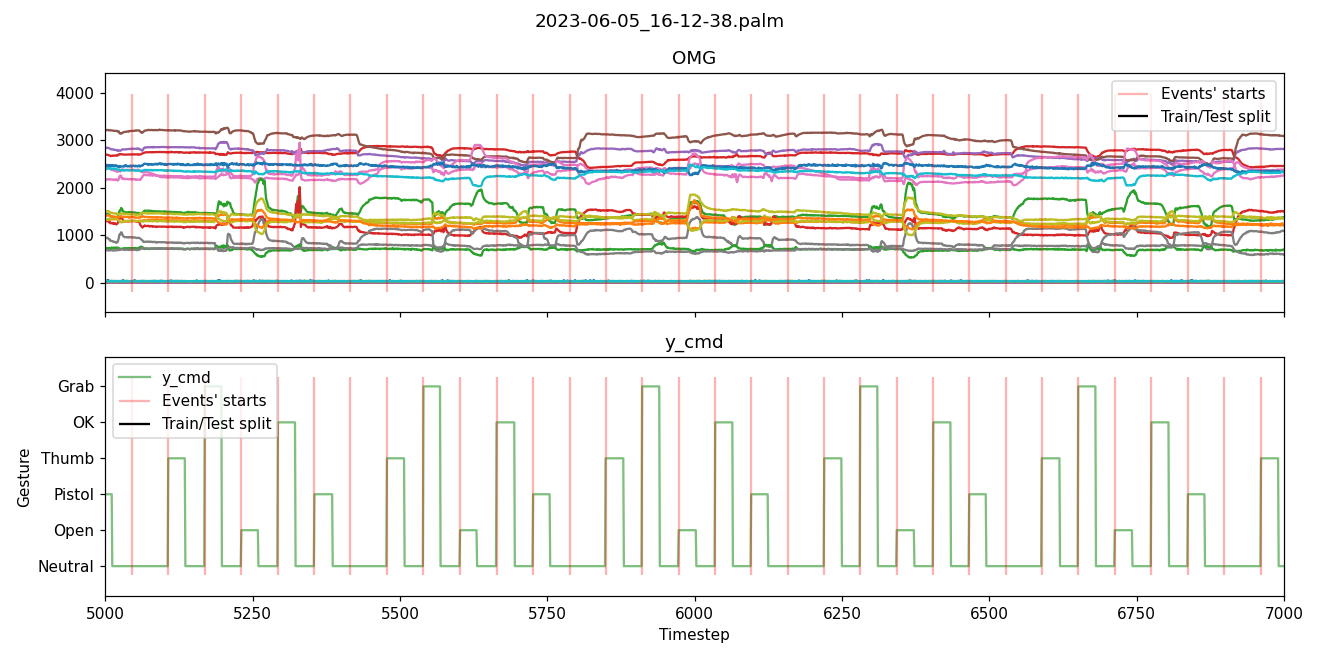
… -> Neutral -> Gesture1 -> Neutral -> Gesture2 -> Neutral -> …

## 

## Пример собранных данных



*Изображение* — верхний график отображает данные, полученные от ОМГ-датчиков. Ось X является осью времени, а ось Y отображает значения датчиков. Нижний график отображает протокол выполнения жестов, которому следовал пилот в ходе сбора данных.

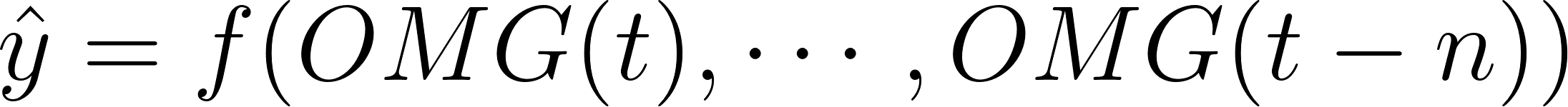


*Изображение* — верхний график отображает ОМГ-данные в приближенном масштабе. Ось X является осью времени, а ось Y отображает значения датчиков. Нижний график отображает протокол выполнения жестов, которому следовал пилот в ходе сбора данных.

Стоит также отметить, что каждому моменту времени соответствует конкретный жест.

# Постановка задачи

Основная задача стажировки заключается в том, чтобы построить модель, которая на основе ОМГ-данных распознавала бы, какой жест выполняется в данный момент. В терминах машинного обучения задача сводится к многоклассовой классификации.

[](https://www.codecogs.com/eqnedit.php?latex=%5Chat%7By%7D%20%3D%20f(OMG(t)%2C%20%5Ccdots%2C%20OMG(t-n))#0)

Участникам стажировки будут представлены наборы данных (данные + протокол) от здоровых и целевых пилотов.

В ходе разработки участниками решений также стоит учитывать следующие ограничения:

1. Скорость инференса решения должна быть меньше 33 мс на пк;

## Репозиторий

[Ссылка](https://github.com/MaxBalashov/motorica-x-skillfactory-gestures/tree/main) на репозиторий с кодом, который может пригодится участникам стажировки.

Также, полезным может оказаться [этот ноутбук](https://github.com/sin-mike/2022_neurotech_course/blob/master/4%20-%20mnist_arduino_example/mnist_arduino_example.ipynb), в котором с помощью MNIST-датасета происходит некоторая симуляция работы с данными типа временных рядов.

## Валидация решений

### Тестовая выборка

В представленных данных есть разделение на обучающую и тестовую выборки. Тестовая выборка является открытой для участников стажировки, на ней можно в оффлайне проверять качество разработанных моделей.

### Онлайн тест

Тестирование с помощью ОМГ-манжеты один из способов проверить разработанные модели в условиях приближенных к целевым. Шаблон инференса модели в режиме реального времени представлен в [ноутбуке](https://github.com/MaxBalashov/motorica-x-skillfactory-gestures/blob/main/baseline_logreg.ipynb) в главе “Real-time inference”.

Предлагается, подставить разработанную модель (а возможно еще и постпроцессинг выхода модели) в данный шаблон. Ноутбук с обучением модели, валидацией на тестовой выборке и инференсом модели в реальном времени на протезе передается Моторике. Сотрудник Моторики проводит валидацию решения участника на пилоте.

# Проблемы\*

Т.е. задачи со звездочкой.

## Задача генерализации монтажей для одного пилота

Монтаж — конкретная установка ОМГ-манжеты на руку пилота. В текущих условиях нельзя дважды установить ОМГ-мажету идентичным образом, поэтому монтажи различаются между собой. Это различие отображается в собранных данных от монтажа к монтажу. Таким образом обученная система принятия решений (СПР) на собранных заранее данных, может быть не готова для работы с данными полученными в ходе установки нового монтажа.

Для продуктового решения необходимо минимизировать время донастройки СПР, так чтобы пилоту не приходилось проходить полный цикл сбора данных в текущем монтаже с последующим полным обучением СПР под этот конкретный монтаж.

## Задача генерализации монтажей для нескольких пилотов

Это обобщение проблемы выше, но для некоторого числа пилотов, по каждому из которых есть один или более монтажей.

Неформальная постановка задачи: “Возможно ли обучить модель на данных некоторого числа пилотов (чем больше тем лучше), которая после обучения будет достаточно хорошо работать на новых данных тех же пилотов.” + можно добавить возможность короткой донастройки СПР с использованием данных текущего монтажа.

## Задача портирования решения на МК

Чтобы разрабатываемые решения были приближены к продуктовым требованиям, стоит учесть, что вычисления необходимые для СПР будут производиться на микроконтроллере. Поэтому решение должно быть потенциально портируемо на микроконтроллер.

Это не означает, что разрешено пользоваться только легкими моделями (деревья решений, логистическая регрессия и тп). Можно использовать и нейронные сети, но есть ограничение на количество операций с плавающей точкой (FLOPS). Например, слои Conv2D имеют слишком большие FLOPS - их вычисление на МК будет занимать много времени. Но даже при таких архитектурах можно воспользоваться различными подходами, например, дистилляцией моделей.