

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
МОСКОВСКИЙ ФИЗИКО-ТЕХНИЧЕСКИЙ ИНСТИТУТ (государственный университет)
ФИЗТЕХ-ШКОЛА ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ
КАФЕДРА «ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ»

Владимиров Эдуард Анатольевич

**Модели пространства состояний в задачах
классификации сигналов ЭКоГ**

010990 — Интеллектуальный анализ данных

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА БАКАЛАВРА

Научный руководитель:
д. ф.-м. н. Стрижов Вадим Викторович

Москва
2023

Содержание

1	Введение	4
2	Обозначения	5
3	Постановка задачи классификации сигнала ЭКоГ	5
4	Обзор существующих подходов к декодированию сигналов	6
5	Модели пространства состояний	6
5.1	Рекуррентные нейронные сети (RNN)	9
5.2	Нейронные контролируемые дифференциальные уравнения (NCDE)	10
5.3	Модель структурированного пространства состояний (S4)	12
5.4	Сравнение моделей	13
6	Вычислительный эксперимент	13
6.1	Экспериментальные данные	14
6.2	Условия проведения эксперимента	14
6.3	Анализ ошибки	14
6.4	Итоговые рекомендации	16
7	Заключение	16
	Список литературы	16

Аннотация

Данная работа посвящена нейронному декодированию — восстановлению стимула по сигналам головного мозга. А именно, рассматривается задача бинарной классификации сигналов, полученных во время движения руки. Для получения высокого качества классификации предлагается использовать модели глубокого обучения. В задаче декодирования сигналов часто применяются свёрточные нейронные сети и трансформеры, в то время как модели пространства состояний применяются редко. Предлагается восполнить данный пробел и подготовить практическое руководство к выбору модели из данного семейства. Проведено сравнение следующих моделей пространства состояний: RNN, NCDE, S4. Для анализа качества алгоритмов предсказания проводится вычислительный эксперимент на данных ЭКоГ.

Ключевые слова: *ЭКоГ, нейронное декодирование, модели пространства состояний, RNN, S4, NCDE*

1 Введение

Нейронное декодирование – процесс расшифровки информации, полученной в результате нейронной активности – имеет большие перспективы в понимании сложностей человеческого мозга и разработке передовых приложений в таких областях, как нейропротезирование и интерфейсы мозг-компьютер. Нейронное декодирование включает в себя извлечение значимой информации из данных об активности головного мозга с целью вывода когнитивных состояний, намерений движения или ощущений. Например, исследователи предсказывают движения, основываясь на активности в моторной коре [1], действия, основанные на активности в префронтальной и теменной коре [2], и пространственные местоположения, основанные на активности в гиппокампе [3]. Расшифровывание сигналов головного мозга даёт нам понять, как мозг обрабатывает и представляет информацию о мире, открывая путь к революционным достижениям в понимании и взаимодействии с мозгом.

По сути, нейронное декодирование – это задача классификации (или регрессии), связывающая нейронные сигналы с определенными переменными. При такой постановке проблемы становится очевидным, что существует широкий спектр методов, которые можно применить. Однако, несмотря на недавние достижения в области методов глубокого обучения, по-прежнему принято декодировать сигналы традиционным подходом. Этот подход включает в себя фильтрацию сигналов от шума, извлечение временных, частотных, пространственных и статистических признаков и применение классического алгоритма машинного обучения, будь то логистическая регрессия, метод опорных векторов или линейный дискриминантный анализ [4]. Использование современных алгоритмов машинного обучения позволяет значительно повысить качество прогноза. Стоит отметить, что модели глубокого обучения применяются не только при прогнозировании, но и для предобработки данных от шума. Для этого используются свёрточные автоэнкодеры [5, 6] и генеративно-состязательные сети [7].

В области нейронного декодирования широкое распространение получили свёрточные нейронные сети [8, 9] и трансформеры [10, 11]. В то же время как модели пространства состояний (SSM) используются исключительно в байесовском ключе, будь то фильтр Калмана [12], скрытые марковские цепи [13] или линейная модель со скрытыми состояниями [14]. То есть, с точки зрения глубокого обучения модели пространства состояний практически не применяются в рассматриваемой задаче. Хотя они дают математическую основу для моделирования сложной динамики нейронных систем. Поэтому, предлагается заполнить этот пробел, применив глубокие нейронные сети, относящиеся к SSM, в задаче нейронного декодирования.

Выбор соответствующей модели является важным, поскольку различные алгоритмы обладают отличительными характеристиками, которые могут значительно влиять на конечный результат. Поэтому крайне важно тщательно изучить и сравнить свойства моделей, чтобы предоставить практическое руководство по их выбору. Понимая сильные и слабые стороны каждой из них, исследователи и практики смогут принимать обоснованные решения при разработке экспериментов и приложений в области нейронного декодирования.

В данной работе рассматриваются следующие модели пространства состояний: рекуррентные нейронные сети (RNN) [15], нейронные контролируемые дифференциальные уравнения (NCDE) [16], модель структурного пространства состояний (S4) [17].

Остальная часть работы организована следующим образом. Раздел 2 раскрывает смысл используемых обозначений. Раздел 3 даёт формальную постановку решаемой задачи. Раздел 4 представляет обзор литературы по применению моделей пространства состояний в задаче нейронного декодирования. Раздел 5 представляет подробный анализ и сравнение моделей RNN, S4 и NCDE. Раздел 6 предоставляет практическое руководство по выбору модели пространства состояний, полученное на основе вычислительного эксперимента. Наконец, раздел 7 завершает работу, подчёркивая её вклад.

2 Обозначения

ЭКоГ — электрокортикограмма

RNN — Recurrent Neural Network

CNN — Convolutional Neural Network

S4 — Structured State Space for Sequence Modeling

NODE/Neural ODE — Neural Ordinary Differential Equation

NCDE/Neural CDE — Neural Controlled Differential Equation

SSM — State Space Models

МКР — межквартильный размах

3 Постановка задачи классификации сигнала ЭКоГ

Пусть $\mathbf{X} \in \mathbb{R}^{M \times N \times T}$ — M измерений ЭКГ, где N — число электродов, T — число элементов временного ряда. Одному измерению ЭКГ соответствуют набор сигналов, записанный на некоторый промежуток времени, во время которого был совершён некий стимул.

$Y \in \{0, 1\}^M$ — целевая переменная, индикатор наличия/отсутствия стимула.

Целевая функция $\mathbf{f} : \mathbf{X} \times W \rightarrow Y$, где $W \in \mathbb{R}^P$ — параметры модели

Критерий качества — бинарная кросс-энтропия с $L2$ -регуляризацией:

$$L(\mathbf{w}) = -\frac{1}{M} \sum_{m=1}^M y_m \log(f(\mathbf{w}, \mathbf{x})) + (1 - y_m) \log(1 - f(\mathbf{w}, \mathbf{x})) + \lambda \|\mathbf{w}\|^2$$

Оптимизационная задача — выбор модели, доставляющей минимум критерия качества:

$$\hat{\mathbf{w}} = \arg \min_{\mathbf{w}} L(\mathbf{w})$$

Основная метрика для этой задачи — точность (ассурасу), поскольку классы сбалансированы.

4 Обзор существующих подходов к декодированию сигналов

Основная задача данной работы — анализ свойств моделей пространства состояний на примере задачи классификации сигналов ЭКоГ на два класса — "движение" и "покой".

На данный момент классический подход выглядит следующим образом: сигнал фильтруется для удаления шумов, затем применяются преобразования для извлечения признаков (среднее и максимум, параметры Хёрша, вейвлет-преобразование, преобразование Фурье или преобразование Гильберта). После этого применяется алгоритм классификации (логистическая регрессия, метод опорных векторов, наивный байесовский классификатор или линейный дискриминантный анализ).

С ростом популярности глубоких нейронных сетей появился новый подход, в котором нейронные сети берут на себя и генерацию признаков, и процесс предсказания. В работе [18] в задаче декодирования движения пальца по данным ЭКоГ использовался гибридный подход: CNN отвечала за извлечение признаков, а RNN отвечал за распознавание временной динамики. В работе [19] также использовалась связка CNN + RNN для определения движения суставов верхней части тела, основанную как на данных ЭКоГ, так и на видеоданных. В работе [20] использовались рекуррентные нейронные сети, которые распознавали временные зависимости в сигналах ЭКоГ для быстрого декодирования жестов.

Таким образом, в глубоких сетях из моделей пространства состояний используется только RNN и её модификации (LSTM, GRU).

Существует ещё один подход — иерархический. В нём объединено несколько алгоритмов и цепочек обработки, которые выполняют переключение или регулировку весов. Применяется в задачах с сигналами, поступающими в реальном времени.

Иллюстрация вышеперечисленных подходов приведена на рисунке 1

5 Модели пространства состояний

Модель пространства состояний — это широкое семейство моделей, которое охватывает целый класс частных случаев, представляющих интерес, во многом такой же, как и линейная регрессия, является модель пространства состояний или динамическая линейная модель, которая была представлена в работах Калмана и Бьюси [12]. Модель возникла в настройках космического слежения, где уравнение состояния определяет уравнения движения для положения или состояния космического аппарата с местоположением x_t , а данные u_t отражают информацию, которую можно наблюдать с устройства слежения, такую как скорость и азимут. Хотя модель была введена как метод, предназначенный главным образом для использования в исследованиях, связанных с аэрокосмической промышленностью, она применялась для моделирования данных из экономики [22, 23, 24] и медицины. Отличной трактовкой анализа временных рядов, основанной на модели пространства состояний, является работа [25]. Современную трактовку нелинейных моделей пространства состояний можно найти в работе [26].

Каноничное представление линейной непрерывной модели пространства состоя-

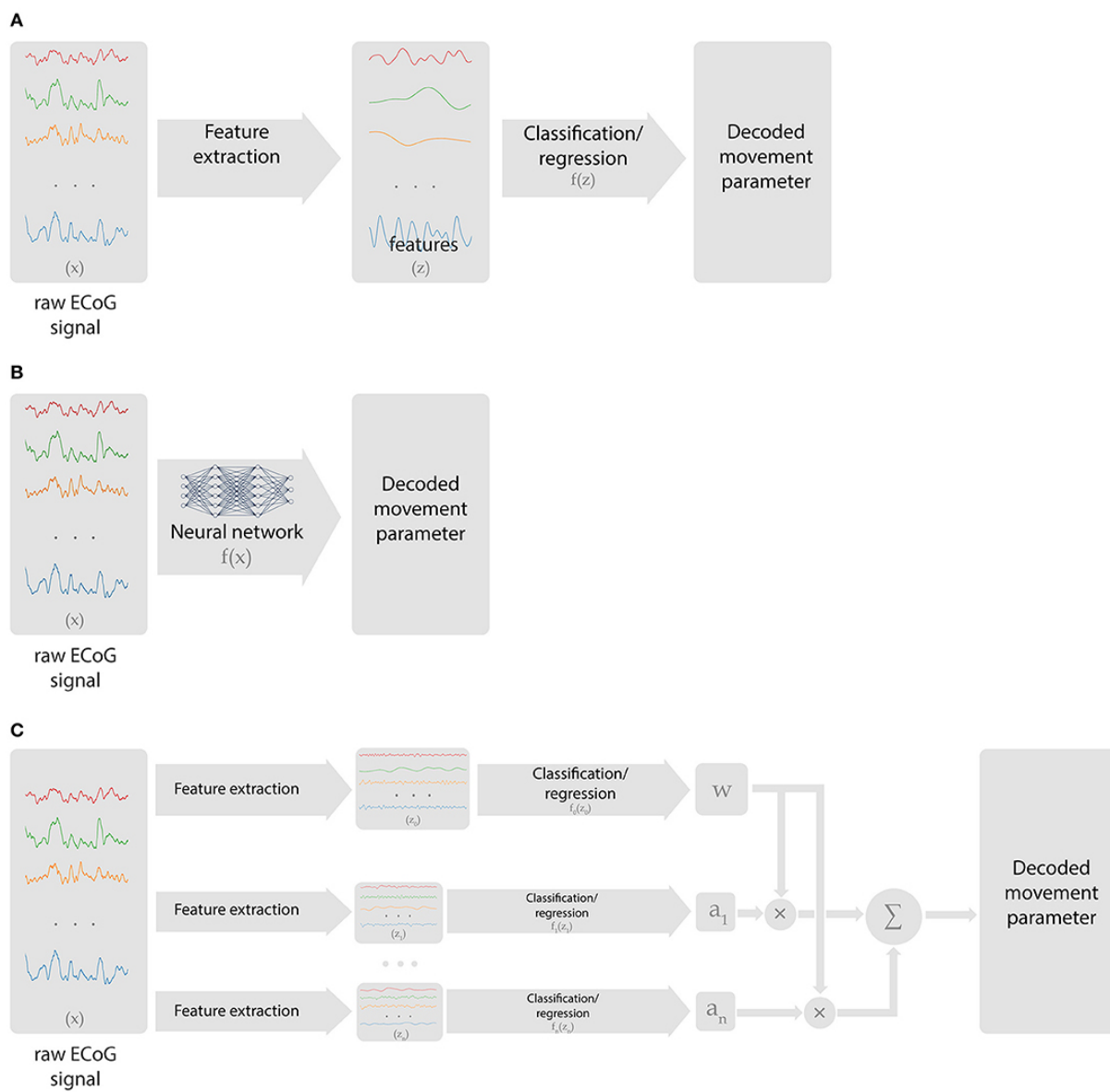


Рис. 1: Подходы к работе с данными ЭКоГ. **(А)** Классический подход. **(В)** Подход глубокого обучения. **(С)** Ансамбль моделей. Взято из [21]

ний таково:

$$\begin{aligned}\mathbf{x}'(t) &= A\mathbf{x}(t) + B\mathbf{u}(t) \\ \mathbf{y}(t) &= C\mathbf{x}(t) + D\mathbf{u}(t),\end{aligned}\tag{1}$$

где $\mathbf{x}(\cdot)$ — вектор состояния, $\mathbf{x}(t) \in \mathbb{R}^K$

$\mathbf{y}(\cdot)$ — вектор выхода, $\mathbf{y}(t) \in \mathbb{R}^s$

$\mathbf{u}(\cdot)$ — вектор входа, $\mathbf{u}(t) \in \mathbb{R}^d$

A — матрица системы, $A \in \mathbb{R}^{K \times K}$

B — матрица входа, $B \in \mathbb{R}^{K \times d}$

C — матрица выхода, $C \in \mathbb{R}^{s \times K}$

D — матрица прямого распространения, $D \in \mathbb{R}^{s \times d}$. По сути, это слагаемое соответствует SkipConnection в сети ResNet [27], и им можно пренебречь. Поэтому далее будем считать $D = 0$.

В силу того, что на практике сигналы имеют дискретное представление, то модель 1 преобразуется в следующий вид:

$$\begin{aligned}\mathbf{x}_k &= A\mathbf{x}_{k-1} + B\mathbf{u}_k \\ \mathbf{y}_k &= C\mathbf{x}_k + D\mathbf{u}_k\end{aligned}\tag{2}$$

Перечислим основные свойства данного семейства моделей:

- + обработка временного ряда любой длины
- + количество параметров не зависит от длины последовательности
- + возможность быстро получить предсказание
- тяжело обучать из-за проблемы взрывающихся/затухающих градиентов и переобучения
- медленное обучение по сравнению со свёрточными нейронными сетями и трансформерами

Далее будут рассмотрены следующие модели пространства состояний: RNN, NCDE и S4. Будут подробно описаны их преимущества и недостатки.

Для дальнейшего анализа моделей введём следующие обозначения:

d — размерность исходного пространства

K — размерность скрытого пространства

s — размерность целевого пространства

$M = \max(d, K, s)$

L — длина обрабатываемой последовательности

5.1 Рекуррентные нейронные сети (RNN)

Рекуррентные нейронные сети являются самой простой и исторически ранней моделью для работы с текстом [15]. В уравнении 3 представлена простая рекуррентная модель — схема Элмана. В дальнейшем были разработаны модификации данной схемы, которые устраняют её недостатки — это модели LSTM и GRU [28, 29], и модель двунаправленной рекуррентной нейронной сети [30].

$$\begin{aligned} \mathbf{x}_t &= \sigma_x(W_x \mathbf{x}_{t-1} + W_u \mathbf{u}_t) \\ \mathbf{y}_t &= \sigma_y(W_y \mathbf{x}_t), \end{aligned} \quad (3)$$

где $\mathbf{u}_i \in \mathbb{R}^d$, $\mathbf{x}_i \in \mathbb{R}^K$, $\mathbf{y}_i \in \mathbb{R}^s$

$\sigma_x : \mathbb{R}^K \rightarrow \mathbb{R}^K$ — функция активации

$\sigma_y : \mathbb{R}^s \rightarrow \mathbb{R}^s$ — функция активации

$W_x \in \mathbb{R}^{K \times K}$, $W_u \in \mathbb{R}^{s \times K}$, $W_y \in \mathbb{R}^{K \times d}$ — матрицы весов

Иллюстрация использования модели RNN в задаче классификации сигналов приведена на рисунке 2.

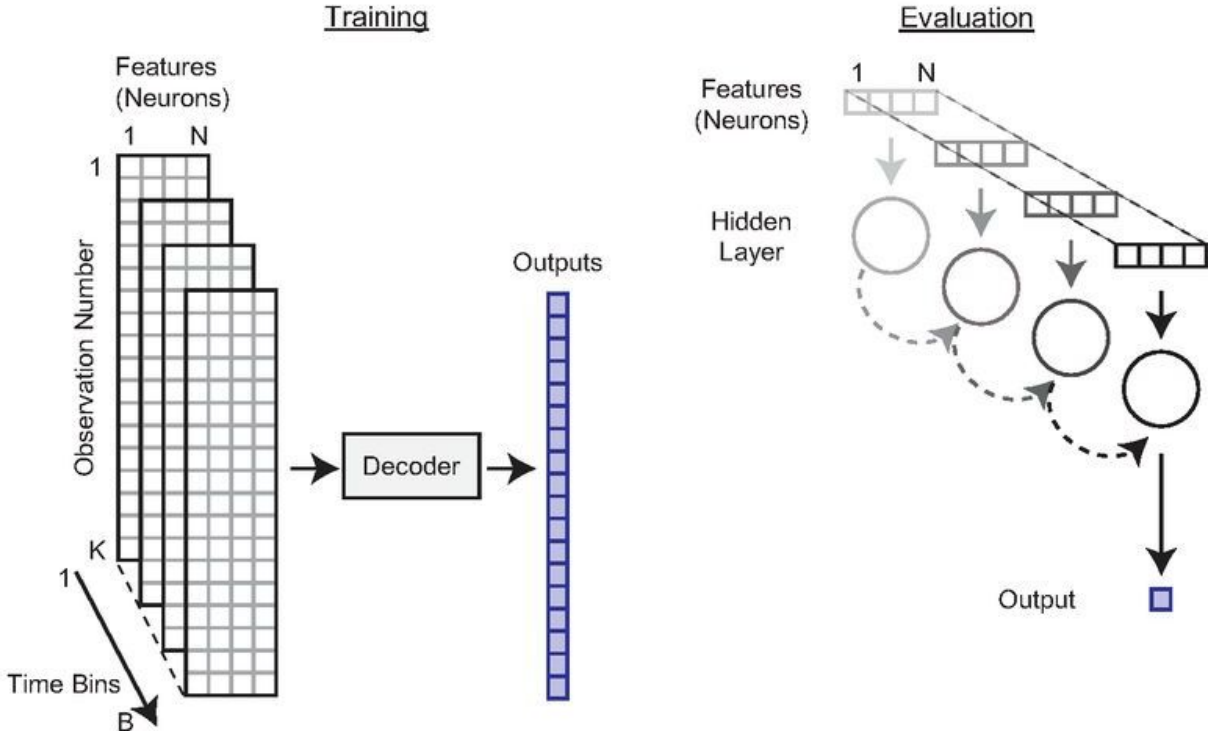


Рис. 2: Схема обучения и предсказания рекуррентного декодера. Взято из [31]

Свойства: количество параметров = $K^2 + Ks + Kd = O(KM)$, время прямого прохода = $O(KML)$

Преимущества: есть реализация во всех фреймворках глубокого обучения, относительно быстрое обучение и предсказание модели

Недостатки: не работают с данными, содержащими пропуски и/или разной частотой сэмпирования

5.2 Нейронные контролируемые дифференциальные уравнения (NCDE)

В модели нейронных обыкновенных дифференциальных уравнений (NODE), прародителе модели NCDE, предполагается, что скрытое состояние описывается дифференциальным уравнением 4.

$$\mathbf{x}'(t) = \mathbf{f}(\mathbf{w}, \mathbf{x}(t)) \quad (4)$$

Данный подход позволяет более качественно описывать временной ряд, порождённый динамической системой. Предсказание модели получается решением дифференциального уравнения 4 с начальными условиями, в котором правая часть задается нейронной сетью. Это означает, что за один прямой проход алгоритм выдает всю траекторию, в отличие от рекуррентных сетей.

Однако у модели NODE есть и минусы. Во-первых, выдаваемое ею решение дифференциального уравнения зависит только от начального состояния, которое постоянно во времени. Во-вторых, не существует механизма для дообучения модели на основе новых данных. Эти пробелы нивелируются моделью нейронных контролируемых дифференциальных уравнений, представленной уравнением 5.

$$\begin{aligned} \mathbf{x}(t_1) &= \zeta(\mathbf{u}_1, t_1) \\ \mathbf{x}(t) &= \mathbf{x}(t_1) + \int_{t_1}^t \mathbf{f}(\mathbf{x}(\tau)) dU(\tau) \\ \mathbf{y}_i &= g(\mathbf{x}(t_i)) \end{aligned} \quad (5)$$

где $\mathbf{u}_i \in \mathbb{R}^d$, $\mathbf{y}_i \in \mathbb{R}^s$

$\mathbf{x} : [t_1, t_L] \rightarrow \mathbb{R}^K$ — функция скрытого состояния

$U : [t_1, t_L] \rightarrow \mathbb{R}^{d+1}$ — кубический сплайн

$\zeta : \mathbb{R}^{d+1} \rightarrow \mathbb{R}^K$ — проектор в скрытое пространство

$f : \mathbb{R}^K \rightarrow \mathbb{R}^{K \times (d+1)}$ — динамика скрытого состояния

$g : \mathbb{R}^K \rightarrow \mathbb{R}^s$ — линейное отображение

Теперь, начальное состояние модели NCDE содержит информацию о времени наблюдения первого объекта. При решении дифференциального уравнения в интегральной форме используется интеграл Римана-Стилтьесса вместо интеграла Римана, агрегируя таким образом все наблюдения. Высокоуровневый план модели NCDE изображён на рисунке 3.

Уделим особое внимание пути $U(t)$, модифицируя который можно получать разные модели:

- $U(t) = t$ — модель Neural ODE [32]
- $U(t) = \sum_{i=1}^{L-1} \mathbf{u}_{t_i} \cdot I(t_i \leq t < t_{i+1})$ — модель ODE-RNN [33]
- $U(t) = \sum_{i=1}^{L-1} \alpha_i(t) \mathbf{u}_{t_i} + (1 - \alpha_i(t)) \mathbf{u}_{t_{i+1}} \cdot I(t_i \leq t < t_{i+1})$, где $\alpha_i(t) = \frac{t_{i+1} - t}{t_{i+1} - t_i}$ — модель с линейной функцией интерполяции

Дообучать модель NCDE можно, решая интегральное уравнение с момента t_L . Однако, данная опция доступна не для всех функций пути: для кубического сплайна

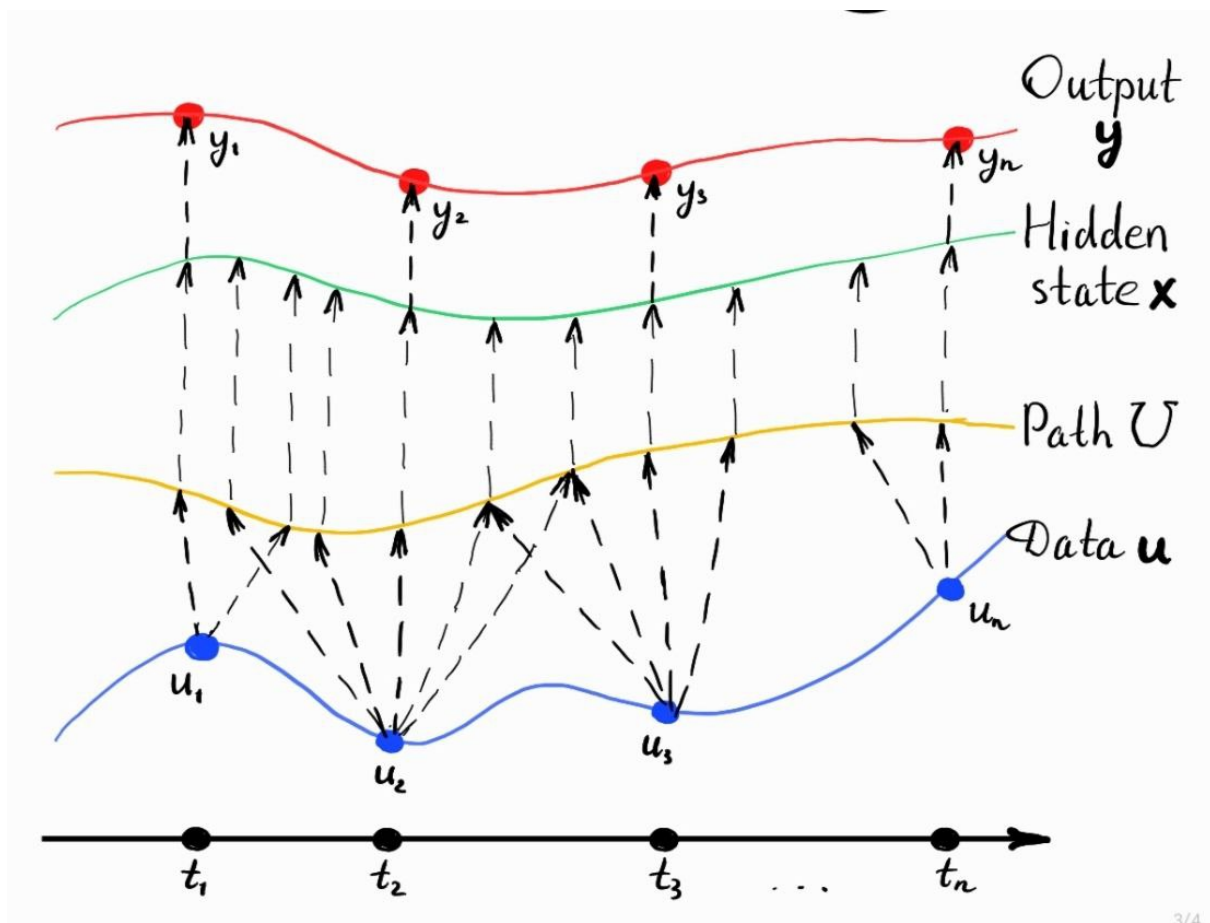


Рис. 3: Иллюстрация алгоритма Neural CDE

— нельзя, а для остальных перечисленных — можно. Связано это с тем, что при получении новых данных, старый путь может оказаться невалиден, вследствие чего невалидным окажется и состояние $\mathbf{x}(t_L)$ [34].

Но и у модели Neural CDE есть недостаток: это большое количество параметров, используемых в функции \mathbf{f} . Если в качестве \mathbf{f} использовать линейную функцию, то число параметров равняется $O(K^2d)$, что на порядок больше, чем у RNN. В оригинальной статье [16] авторы использовали линейную функцию с малым K . Также можно использовать малоранговое линейное преобразование, использующее $O(K^2 + Kd)$ параметров, но в статье [16] показана неэффективность данного подхода.

Свойства: количество параметров = $O(K^2d + Ks)$, время прямого прохода = $O(K^2dL)$

Преимущества: работает с данными, в которых содержатся пропуски и которые имеют разную частоту сэмпирования, большая гибкость в настройке модели: выбор архитектуры функции $\mathbf{f}(\mathbf{x})$ и пути $U(t)$

Недостатки: в десятки раз медленнее, чем RNN, на текущий момент не существует эффективной реализации, долгая настройка модели, нестабильное обучение при больших K , при использовании солверов с адаптивным шагом и функции активации ReLU

5.3 Модель структурированного пространства состояний (S4)

Данная модель строится из модели пространства состояний в 3 шага. Во-первых, применяется билинейное преобразование для перехода из непрерывной постановки 1 в дискретную 2 со следующими характеристиками:

$$\begin{aligned} u_i &= u(i\Delta) \\ \bar{A} &= (I - \frac{\Delta}{2}A)^{-1}(I + \frac{\Delta}{2}A) \\ \bar{B} &= (I - \frac{\Delta}{2}A)^{-1}\Delta B \\ \bar{C} &= C \\ \bar{D} &= 0 \end{aligned}$$

где Δ — шаг дискретизации.

Далее переходим к свёрточному представлению модели 2 с вышеуказанными параметрами. Делается это из соображений эффективности по времени.

$$\begin{aligned} \mathbf{x}_0 &= \bar{B}\mathbf{u}_0 & \mathbf{y}_0 &= \bar{C}\bar{B}\mathbf{u}_0 \\ \mathbf{x}_1 &= \bar{A}\bar{B}\mathbf{u}_0 + \bar{B}\mathbf{u}_1 & \mathbf{y}_1 &= \bar{C}\bar{A}\bar{B}\mathbf{u}_0 + \bar{C}\bar{B}\mathbf{u}_1 \\ \mathbf{x}_2 &= \bar{A}^2\bar{B}\mathbf{u}_0 + \bar{A}\bar{B}\mathbf{u}_1 + \bar{B}\mathbf{u}_2 & \mathbf{y}_2 &= \bar{C}\bar{A}^2\bar{B}\mathbf{u}_0 + \bar{C}\bar{A}\bar{B}\mathbf{u}_1 + \bar{C}\bar{B}\mathbf{u}_2 \\ \dots & & \dots & \end{aligned}$$

$$\begin{aligned} \mathbf{y}_i &= \bar{C}\bar{A}^i\bar{B}\mathbf{u}_0 + \bar{C}\bar{A}^{i-1}\bar{B}\mathbf{u}_1 + \dots + \bar{C}\bar{A}\bar{B}\mathbf{u}_{i-1} + \bar{C}\bar{B}\mathbf{u}_i \\ \mathbf{y} &= \bar{\mathbf{K}} * \mathbf{u}, \text{ где } \bar{\mathbf{K}} = (\bar{C}\bar{B}, \bar{C}\bar{A}\bar{B}, \dots, \bar{C}\bar{A}^{L-1}\bar{B}) \end{aligned}$$

Таким образом, если ядро свёртки $\bar{\mathbf{K}}$ уже предсчитано, то можно быстро применить данный слой. Однако для матрицы A общего вида вычисление займёт $O(LK^2)$

памяти и $O(LK^3)$ времени (при $d = s = 1$). Для экономии памяти вместо ядра свёртки хранятся значения производящей функции от данной последовательности в корнях степени L от 1. Для уменьшения же временных затрат на матрицу A накладывают ограничения — она должна иметь следующий вид:

$$A = \Lambda - PQ^*,$$

где $\Lambda \in \mathbb{R}^{K \times K}$ — диагональная матрица, $P, Q \in \mathbb{R}^{K \times 1}$

И в третьих, с помощью инициализации HIPPO получаются начальные приближения для Λ, P, Q . Инициализация HIPPO имеет следующий вид:

$$A_{nk} = - \begin{cases} \sqrt{(2n+1)(2k+1)} & \text{при } n > k \\ n+1 & \text{при } n = k \\ 0 & \text{при } n < k \end{cases}$$

Подробную информацию о каждом шаге можно найти в работе [17].

Свойства: количество параметров = $Kd + Ks$, время прямого прохода = $O(KML)$

Преимущества: хорошо работает с данными, содержащими долговременные зависимости, есть эффективная реализация [35]

Недостатки: в несколько раз медленнее, чем RNN, не работают с данными, содержащими пропуски и/или разной частотой сэмпирования

5.4 Сравнение моделей

В таблице 1 приведено описание свойств различных моделей глубокого обучения. Помимо моделей пространства состояний, для полноты картины указаны свёрточные нейронные сети и трансформеры.

Для CNN используются другие обозначения: K — размер ядра свёртки, Ch — количество каналов ЭКоГ. Считаем, что количество каналов в свёртке равняется одному и что размер ядра свёртки составляет $K \times K$

Таблица 1: Сравнение моделей по числу обучаемых параметров, времени прямого прохода, наличию эффективной реализации, возможности работы с пропусками и хранения всей истории

	Parameters	Forward	Fast impl.	Cont. time	Unb. context
RNN	$O(KM)$	$O(KML)$	+	—	+
NCDE	$O(K^2d + Ks)$	$O(K^2dL)$	—	+	+
S4	$O(Kd + Ks)$	$O(KML)$	+	—	+
Transformer	$O(K^2d + Ks)$	$O(LK \cdot (L + s))$	+	—	+
CNN	$O(K^2)$	$O(K^2(L - K)(N - K))$	+	—	—

6 Вычислительный эксперимент

Целью эксперимента является сравнение ранее перечисленных методов пространства состояний. Эти методы применяются для предсказания наличия стимула по сигналам головного мозга.

6.1 Экспериментальные данные

Одновременные ЭКоГ-записи были получены от 12 участников (8 мужчин, 4 женщины) в ходе непрерывного клинического мониторинга эпилепсии, проводимого в медицинском центре Харборвью в Сиэтле. Эти записи длятся 7 ± 2 дня на каждого участника. Возраст участников составляет 29 ± 8 лет, и им были имплантированы электроды, преимущественно в одно полушарие (5 правых, 7 левых).

Задача декодирования заключается в том, чтобы классифицировать события "движения" и "покоя" верхней конечности руки, противоположной полушарию имплантированного электрода. События перемещения соответствуют движению запястья, которое произошло по крайней мере через 0.5 с без движения, в то время как события покоя указывают на отсутствие движения ни в одном запястье в течение по крайней мере 3 с.

Обработка данных ECoG осуществлена с использованием обычных скриптов MNE-Python. Сначала был удалён средний дрейф постоянного тока и высокоамплитудные разрывы. Затем данные ECoG каждого участника подвергались полосовой фильтрации (1-200 Гц), notch-фильтрации и повторной привязке к общей медиане по электродам. Также удалены зашумлённые сигналы на основе аномального стандартного отклонения (> 5 МКР) или эксцесса (> 10 МКР). Затем сгенерированы 10-секундные сегменты ECoG, сосредоточенные вокруг каждого события "движение" и "отдых". Сегменты ECoG с отсутствующими данными или большими артефактами были удалены на основе аномальной спектральной плотности мощности [36]. Затем частота сигнала была уменьшена до 250 Гц, а временные интервалы сокращены до 2 секунд по центру каждого события. Для каждого участника было сбалансировано количество сегментов движения и отдыха в течение каждого дня записи, в результате чего на одного участника пришлось 1155 ± 568 событий.

6.2 Условия проведения эксперимента

Структура модели декодирования сигнала ЭКоГ приведена на рисунке 4. Вначале применяется модель пространства состояний, затем идёт агрегирующий слой, который по каждому элементу и каналу берёт максимум, среднее и последний элемент, после идёт функция активации ReLU и линейное преобразование.

Ввиду переобучения моделей, в Seq.model внедрён SpatialDropout, который за-нуляет сразу весь канал, а не его случайные элементы.

Участники разбиваются на обучающую, валидационную и тестовую выборки в соотношении 7 – 4 – 1. Модель обучается 20 раз, каждая на одном из 20 разбиений, результаты предсказания затем усредняются.

В качестве рекуррентной нейронной сети используется LSTM, в Neural CDE используются кубические сплайны с методом Рунге-Кутты 4го порядка.

Константы: $\lambda = 0.002$ (L2-регуляризация), $p = 0.5$ (dropout), batch size = 32, hidden size = 32, оптимизатор = AdamW с lr = 0.002,

6.3 Анализ ошибки

Результаты обучения моделей показаны на рисунке 5 и в таблице 2. Видим, что ассигасу всех моделей находится возле 0.5, с учётом равномерного распределения

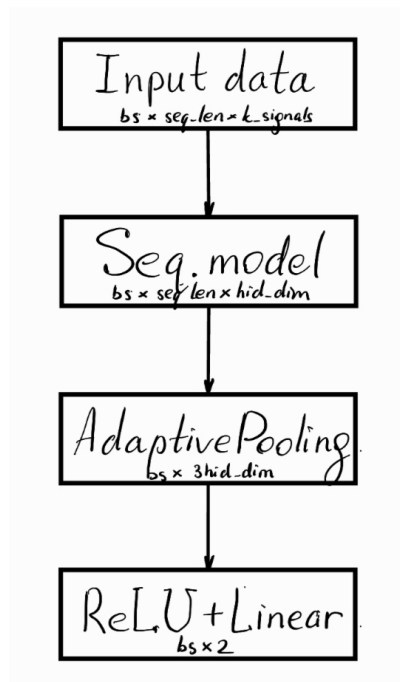


Рис. 4: Архитектура модели классификации

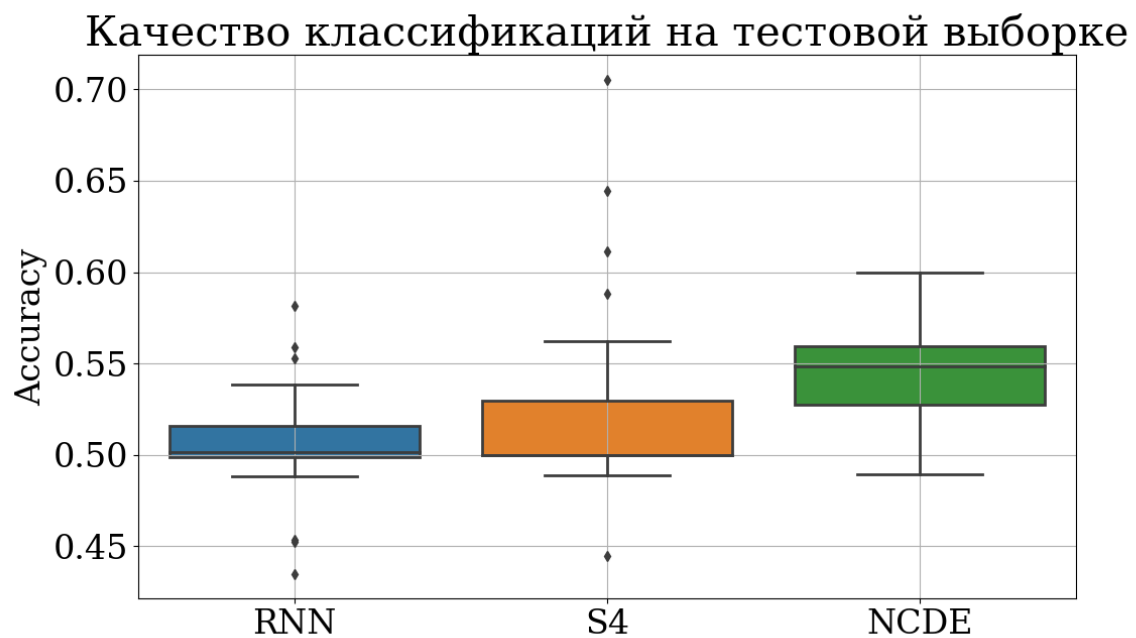


Рис. 5: Точность предсказания моделей пространства состояний

классов это говорит о низком качестве предсказания. Связано это с тем, что модель не выучила "хорошее" признаковое представление сигналов. В таких случаях принято использовать CNN для извлечения "полезных" признаков и увеличения качества прогноза. Исходя из полученных результатов, можно заключить, что Neural CDE имеет лучшую точность предсказания на тестовой выборке. Но время обучения превышает время RNN почти в 8 раз.

Таблица 2: Сравнение характеристик RNN, S4 и NCDE

	Parameters	Time per epoch (sec)	Accuracy
RNN	34.2k	5.86	0.506 ± 0.027
S4	33.3k	10.07	0.521 ± 0.049
Neural CDE	31.5k	37.23	0.546 ± 0.026

6.4 Итоговые рекомендации

Предлагается использовать модели семейства RNN в качестве начального подхода. Далее, если длина последовательности слишком большая (порядка 10 тысяч) или если требуется получить более высокое качество предсказания, то стоит приглядеться к модели S4. Затем, если в данных есть пропущенные значения, а при текущем методе заполнения пропусков (например, средним или константой) качество предсказания по-прежнему невысокое, или если частота сэмплирования непостоянна во времени, то нужно применить модель Neural CDE. Отдельного внимания заслуживает выбор функции пути. В случае, если необходимо делать прогнозы в режиме онлайн, то подойдут прямолинейная и линейная интерполяции. Иначе, стоит выбрать кубические сплайны. Однако нужно учесть, что время обучения данной модели в десятки раз больше, чем у рекуррентных нейронных сетей. Также случаются провалы в работе модели с настройками по умолчанию, так что придётся уменьшать максимальное значение ошибки, менять решатель дифференциального уравнения и архитектуру модели.

7 Заключение

Рассмотрено применение моделей пространства состояний в задаче классификации сигналов ЭКоГ. Изучены свойства следующих моделей пространства состояний: RNN, NCDE, S4. Предоставлена рекомендация по выбору моделей пространства состояния. Продемонстрировано, что модель Neural CDE имеет лучшее качество на тестовой выборке по сравнению с другими моделями, однако её время обучения сильно превышает время обучения других моделей.

Список литературы

- [1] Christian Ethier, Emily R Oby, Matthew J Bauman, and Lee E Miller. Restoration of grasp following paralysis through brain-controlled stimulation of muscles. *Nature*, 485(7398):368–371, 2012.

- [2] Guilhem Ibos and David J Freedman. Sequential sensory and decision processing in posterior parietal cortex. *Elife*, 6:e23743, 2017.
- [3] Thomas J Davidson, Fabian Kloosterman, and Matthew A Wilson. Hippocampal replay of extended experience. *Neuron*, 63(4):497–507, 2009.
- [4] Rezwana Firuzi, Hamed Ahmadyani, Mohammad Foad Abdi, Dana Naderi, Jahan Hassan, and Ayub Bokani. Decoding neural signals with computational models: A systematic review of invasive bmi. *arXiv preprint arXiv:2211.03324*, 2022.
- [5] Niago Moreira Nobre Leite, Eanes Torres Pereira, Edmar Candeia Gurjao, and Luciana Ribeiro Veloso. Deep convolutional autoencoder for eeg noise filtering. In *2018 IEEE international conference on bioinformatics and biomedicine (BIBM)*, pages 2605–2612. IEEE, 2018.
- [6] Arthur Sena Lins Caldas, Eanes Torres Pereira, Niago Moreira Nobre Leite, Arthur Dimitri Brito Oliveira, and Ellen Ribeiro Lucena. Towards automatic eeg signal denoising by quality metric optimization. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7. IEEE, 2020.
- [7] Yang An, Hak Keung Lam, and Sai Ho Ling. Auto-denoising for eeg signals using generative adversarial network. *Sensors*, 22(5):1750, 2022.
- [8] Vernon J Lawhern, Amelia J Solon, Nicholas R Waytowich, Stephen M Gordon, Chou P Hung, and Brent J Lance. Eegnet: a compact convolutional neural network for eeg-based brain–computer interfaces. *Journal of neural engineering*, 15(5):056013, 2018.
- [9] Steven M Peterson, Zoe Steine-Hanson, Nathan Davis, Rajesh PN Rao, and Bingni W Brunton. Generalized neural decoders for transfer learning across participants and recording modalities. *Journal of Neural Engineering*, 18(2):026014, 2021.
- [10] Alexandre Défossez, Charlotte Caucheteux, Jérémy Rapin, Ori Kabeli, and Jean-Rémi King. Decoding speech from non-invasive brain recordings. *arXiv preprint arXiv:2208.12266*, 2022.
- [11] Jerry Tang, Amanda LeBel, Shailee Jain, and Alexander G Huth. Semantic reconstruction of continuous language from non-invasive brain recordings. *Nature Neuroscience*, pages 1–9, 2023.
- [12] Rudolph E Kalman and Richard S Bucy. New results in linear filtering and prediction theory. 1961.
- [13] Liam Paninski, Yashar Ahmadian, Daniel Gil Ferreira, Shinsuke Koyama, Kamiar Rahnama Rad, Michael Vidne, Joshua Vogelstein, and Wei Wu. A new look at state-space models for neural data. *Journal of computational neuroscience*, 29:107–126, 2010.
- [14] Wei Wu, Jayant E Kulkarni, Nicholas G Hatsopoulos, and Liam Paninski. Neural decoding of hand motion using a linear state-space model with hidden states. *IEEE Transactions on neural systems and rehabilitation engineering*, 17(4):370–378, 2009.

- [15] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science, 1985.
- [16] Patrick Kidger, James Morrill, James Foster, and Terry Lyons. Neural controlled differential equations for irregular time series. *Advances in Neural Information Processing Systems*, 33:6696–6707, 2020.
- [17] Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. *arXiv preprint arXiv:2111.00396*, 2021.
- [18] Ziqian Xie, Odelia Schwartz, and Abhishek Prasad. Decoding of finger trajectory from ecog using deep learning. *Journal of neural engineering*, 15(3):036009, 2018.
- [19] Nancy Wang, Ali Farhadi, Rajesh Rao, and Bingni Brunton. Agile movement prediction: Multimodal deep learning for natural human neural recordings and video. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.
- [20] Anming Du, Shuqin Yang, Weijia Liu, and Haiping Huang. Decoding ecog signal with deep learning model based on lstm. In *TENCON 2018-2018 IEEE Region 10 Conference*, pages 0430–0435. IEEE, 2018.
- [21] Ksenia Volkova, Mikhail A Lebedev, Alexander Kaplan, and Alexei Ossadtchi. Decoding movement from electrocorticographic activity: a review. *Frontiers in neuroinformatics*, 13:74, 2019.
- [22] P Jeffrey Harrison and Colin F Stevens. Bayesian forecasting. *Journal of the Royal Statistical Society: Series B (Methodological)*, 38(3):205–228, 1976.
- [23] Andrew C Harvey and Richard G Pierse. Estimating missing observations in economic time series. *Journal of the American statistical Association*, 79(385):125–131, 1984.
- [24] Andrew C Harvey and Paul HJ Todd. Forecasting economic time series with structural and box-jenkins models: A case study. *Journal of Business & Economic Statistics*, 1(4):299–307, 1983.
- [25] James Durbin and Siem Jan Koopman. A simple and efficient simulation smoother for state space time series analysis. *Biometrika*, 89(3):603–616, 2002.
- [26] Randal Douc, Eric Moulines, and David Stoffer. *Nonlinear time series: Theory, methods and applications with R examples*. CRC press, 2014.
- [27] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [28] Klaus Greff, Rupesh K Srivastava, Jan Koutník, Bas R Steunebrink, and Jürgen Schmidhuber. Lstm: A search space odyssey. *IEEE transactions on neural networks and learning systems*, 28(10):2222–2232, 2016.

- [29] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.
- [30] Alex Graves, Santiago Fernández, and Jürgen Schmidhuber. Bidirectional lstm networks for improved phoneme classification and recognition. In *Artificial Neural Networks: Formal Models and Their Applications–ICANN 2005: 15th International Conference, Warsaw, Poland, September 11-15, 2005. Proceedings, Part II 15*, pages 799–804. Springer, 2005.
- [31] Joshua I Glaser, Ari S Benjamin, Rameed H Chowdhury, Matthew G Perich, Lee E Miller, and Konrad P Kording. Machine learning for neural decoding. *Eneuro*, 7(4), 2020.
- [32] Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.
- [33] Yulia Rubanova, Ricky TQ Chen, and David K Duvenaud. Latent ordinary differential equations for irregularly-sampled time series. *Advances in neural information processing systems*, 32, 2019.
- [34] James Morrill, Patrick Kidger, Lingyi Yang, and Terry Lyons. Neural controlled differential equations for online prediction tasks. *arXiv preprint arXiv:2106.11028*, 2021.
- [35] HazyResearch. state-spaces. <https://github.com/HazyResearch/state-spaces>, 2021.
- [36] Steven M Peterson, Satpreet H Singh, Nancy XR Wang, Rajesh PN Rao, and Bingni W Brunton. Behavioral and neural variability of naturalistic arm movements. *Eneuro*, 8(3), 2021.