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

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 заполнить 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*

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 машинного обучения, по-прежнему принято декодировать активность традиционными методами, такими как логистическая регрессия. Использование современных инструментов ML для нейронного декодирования значительно повышает качество прогноза и глубже проникнуть в функции нервной системы Стоит отметить, что модели машинного обучения применяются не только при прогнозировании, но и для предобработки данных от шума. Для этого используются свёрточные автоэнкодеры [4, 5], фильтр Калмана [6] и генеративно-состязательные сети [7].

Модели машинного обучения, особенно модели пространства состояний, дают математическую основу для моделирования сложной динамики нейронных систем. Модели пространства состояний предлагают гибкий подход, представляя нейронную активность как последовательность скрытых состояний, развивающихся со временем, что обеспечивает структурированное представление, упрощающее извлечение значимой информации.

Используются скрытые марковские цепи [6], линейная модель пространства состояний со скрытыми состояниями [8]

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

В данной работе сравниваются следующие модели пространства состояний: рекуррентные нейронные сети (RNN), контролируемые дифференциальные уравнения (NCDE) [9], модель структурного пространства состояний (S4) [10]

Остальная часть работы организована следующим образом: TODO.

Обозначения

- ЭКоГ — электрокортикограмма
- RNN — Recurrent Neural Network
- S4 — Structured State Space for Sequence Modeling
- NODE — Neural Ordinary Differential Equation
- NCDE — Neural Controlled Differential Equation
- МКР — межквартильный размах

1 Постановка задачи

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 $\mathbf{f} : \mathbf{X} \times W \to Y$, 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 (accuracy), поскольку классы сбалансированы.

2 Обзор существующих алгоритмов классификации

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

На данный момент решение задачи выглядит следующим образом: сигнал фильтруется для удаления шумов, затем применяются преобразования для извлечения признаков (среднее и максимум, параметры Хёрша, вейвлет-преобразование или преобразование Фурье или преобразование Гильберта). После этого применяется алгоритм классификации.

TODO: краткий обзор методов извлечения признаков

1. LR, логистическая регрессия [**?**]

   Простая линейная модель для решения задачи классификации с L2-регуляризацией. В данном классификаторе варьируется константа регуляризации.

2. SVM, классификатор на методе опорных векторов [**?**].

   Модель для классификации с L2-регуляризацией. Алгоритм находит опорные векторы каждого класса, максимизируя расстояние между ними. Гиперплоскость между опорными векторами — граница принятия решения.
   В качестве ядра рассматриваются линейная и радиально-базисная функции. Также варьируется константа регуляризации.

3. LDA, линейный дискриминантный анализ.

   Линейная классификация в пердположениях, что каждый класс распределён нормально и все классы имеют одинаковую матрицу ковариации.[**?**]

4. ERPCov TS LR

   По данным ЭКоГ строятся матрицы ковариаций событийных потенциалов (event-related potentials, ERP) [**?**]. Полученные матрицы проецируются в касательное пространство (TS) [**?**], после чего проводится классификация с помощью логистической регрессии.
   Варьируются тип оценочной функции для ERP ковариаций и константа регуляризации в логистической регрессии.

3 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 [11]. 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 [12, 13, 14] и медицины. Отличной трактовкой анализа временных рядов, основанной на модели пространства состояний, является работа [15]. Современную трактовку нелинейных моделей пространства состояний можно найти в работе [16].

Каноничное представление линейной непрерывной модели пространства состояний 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 [17], 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

3.1 Рекуррентные нейронные сети (RNN)

Рекуррентные нейронные сети являются самой простой и исторически ранней моделью для работы с текстом [18]. 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 [19, 20], и модель двунаправленной рекуррентной нейронной сети [21].

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

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

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

$\sigma_y : \mathbb{R}^s \to \mathbb{R}^s$ — 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

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 сэмплирования

3.2 Нейронные контролируемые дифференциальные уравнения (NCDE)

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

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

Данный подход позволяет более качественно описывать временной ряд, порождённый динамической системой. Предсказание модели получаются решением дифференциально- го 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

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

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

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

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

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

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

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

Теперь, начальное состояние модели NCDE содержит информацию о времени наблюдения первого объекта. При решении дифференциального уравнения в интегральной форме используется интеграл Римана-Стильтьесса вместо интеграла Римана, агрегируя таким образом все наблюдения

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

- $U(t) = t$ — модель Neural ODE [22]
- $U(t) = \sum_{i=1}^{L-1} \mathbf{u}_{t_i} \cdot I(t_i \leqslant t < t_{i+1})$ — модель ODE-RNN [23]
- $U(t) = \sum_{i=1}^{L-1} \alpha_i(t)\mathbf{u}_{t_i} + (1 - \alpha_i(t))\mathbf{u}_{t_{i+1}} \cdot I(t_i \leqslant t < t_{i+1})$, где $\alpha_i(t) = \dfrac{t_{i+1} - t}{t_{i+1} - t_i}$ — 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

— 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 $\mathbf{x}(t_L)$ [24].

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 [9] авторы использовали линейную функцию с малым K. Также можно использовать малоранговое линейное преобразование, использующее $O(K^2 + Kd)$ параметров, но в статье [9] 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 сэмплирования
- 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

3.3 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:

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

где Δ — шаг дискретизации.

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

$$
\begin{aligned}
&\mathbf{x}_0 = \overline{B}\mathbf{u}_0 && \mathbf{y}_0 = \overline{CB}\mathbf{u}_0 \\
&\mathbf{x}_1 = \overline{AB}\mathbf{u}_0 + \overline{B}\mathbf{u}_1 && \mathbf{y}_1 = \overline{CAB}\mathbf{u}_0 + \overline{CB}\mathbf{u}_1 \\
&\mathbf{x}_2 = \overline{A}^2\overline{B}\mathbf{u}_0 + \overline{AB}\mathbf{u}_1 + \overline{B}\mathbf{u}_2 && \mathbf{y}_2 = \overline{CA}^2\overline{B}\mathbf{u}_0 + \overline{CAB}\mathbf{u}_1 + \overline{CB}\mathbf{u}_2 \\
&\ldots && \ldots
\end{aligned}
$$

$$
\mathbf{y}_i = \overline{CA}^i\overline{B}\mathbf{u}_0 + \overline{CA}^{i-1}\overline{B}\mathbf{u}_1 + \ldots + \overline{CAB}\mathbf{u}_{i-1} + \overline{CB}\mathbf{u}_i
$$
$$
\mathbf{y} = \overline{\mathbf{K}} * \mathbf{u},\ \text{где } \overline{\mathbf{K}} = (\overline{CB}, \overline{CAB}, \ldots, \overline{CA}^{L-1}\overline{B})
$$

Таким образом, если ядро свёртки $\overline{\mathbf{K}}$ уже предпосчитано, 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 [10].

Свойства:

- количество параметров $= Kd + Ks$
- время прямого прохода $= O(KML)$

Преимущества:

- хорошо работает с данными, содержащими долговременные зависимости
- есть эффективная реализация [25]

Недостатки:

- в несколько раз медленнее, чем RNN
- не работают с данными, содержащими пропуски и/или разной частотой сэмплирования

3.4 Сравнение моделей

TODO: написать затравку

	Parameters	Forward
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	TODO	TODO

4 Вычислительный эксперимент

4.1 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 сгенерированны 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 [26]. 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.

4.2 Результаты

TOADD

Заключение

Показано, что рекуррентные нейронные сети, нейронные контролируемые дифференциальные уравнения и модель S4 являются частными случаями модели пространства состояний.

Продемонстрировано, что NCDE 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] 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.

[5] 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.

[6] 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.

[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] 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.

[9] 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.

[10] Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. *arXiv preprint arXiv:2111.00396*, 2021.

[11] Rudolph E Kalman and Richard S Bucy. New results in linear filtering and prediction theory. 1961.

[12] P Jeffrey Harrison and Colin F Stevens. Bayesian forecasting. *Journal of the Royal Statistical Society: Series B (Methodological)*, 38(3):205–228, 1976.

[13] 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.

[14] 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.

[15] James Durbin and Siem Jan Koopman. A simple and efficient simulation smoother for state space time series analysis. *Biometrika*, 89(3):603–616, 2002.

[16] Randal Douc, Eric Moulines, and David Stoffer. *Nonlinear time series: Theory, methods and applications with R examples*. CRC press, 2014.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.

[23] 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.

[24] James Morrill, Patrick Kidger, Lingyi Yang, and Terry Lyons. Neural controlled differential equations for online prediction tasks. *arXiv preprint arXiv:2106.11028*, 2021.

[25] HazyResearch. state-spaces. https://github.com/HazyResearch/state-spaces, 2021.

[26] 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.