Санкт-Петербургский государственный университет
Факультет прикладной математики – процессов управления

Беляев Василий Андреевич

Магистерская диссертация

Нейросетевые вычисления динамических параметров

01.04.02 – Прикладная математика и информатика
Математическое моделирование в задачах естествознания

Научный руководитель,
кандидат физ.-мат. наук,
доцент
Степенко Н. А.

Рецензент,
ООО "О.Г.С. Руссия
математик-программист
кандидат физ.-мат. наук,
Купцов С. Ю.

Санкт-Петербург
2020

Содержание

Введение

Нейронная сеть — это тип машинного обучения, который моделирует себя в качестве человеческого мозга. Это создает искусственную нейронную сеть, которая с помощью алгоритма позволяет компьютеру учиться путем включения новых данных. Хотя в наши дни существует множество алгоритмов искусственного интеллекта, нейронные сети способны выполнять то, что называют глубоким обучением. В то время как основной единицей мозга является нейрон, основным строительным блоком искусственной нейронной сети является персептрон, который выполняет простую обработку сигналов, и они затем соединяются в большую ячеистую сеть. Компьютер с нейронной сетью учат выполнять задачу, анализируя обучающие примеры, которые были предварительно подготовлены. Типичным примером задачи для нейронной сети, использующее глубокое обучение, является задача распознавания объектов, где нейронная сеть представлена большим количеством объектов определенного типа, таких как кошка или дорожный знак, и компьютер, анализируя повторяющиеся шаблоны в представленных изображениях, учится классифицировать новые изображения.

В свою очередь, под рекуррентными сетями подразумеваются нейронные сети, в которых присутствуют направленные последовательности при соединении элементов. Это их свойство даёт возможность обрабатывать серии событий во времени или в последовательных цепях пространства. Рекурсивные сети способны использовать внутреннюю память при обработке последовательностей произвольной длины, что неспособны делать в большинстве своём многослойные перцептроны. Как итог, сети RNN применимы в задачах, требующих деления чего-то интегрального на части: распознавание рукописного ввода или речи. Для этих сетей придумано большое количество алгоритмов различной сложности. В последние годы наиболее широко используется сеть с долговременной и кратковременной памятью (LSTM) и контролируемой рекуррентной единицей (GRU).

В качестве примера рассмотрим простейшую нейросеть — перцептрон. Он представляет собой один слой нейронов, принимающих входные данные (один или несколько битов, действительных чисел, пикселей и т.п.), модифицирующих их с учетом собственного веса и передающих далее. В однослойном перцептроне выдача всех нейронов объединяется различными

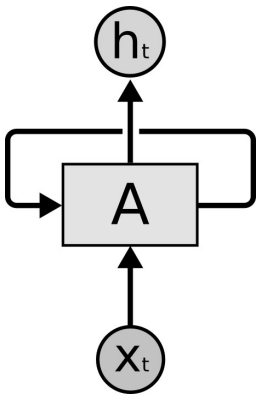


Рис. 1: Модель рекуррентной нейронной сети

способами, после чего от нейросети получается ответ, однако возможности данной конструкции весьма ограничены. При необходимости получения более расширенного функционала, можно пойти другими путями путями, к примеру, прибавляя дополнительное количество слоев и вставить операцию свертки, которая разделяла бы входящие данные на 35 «кусков» разного масштаба. В таком случае на выходе мы имеем сверхточные нейросети для глубинного обучения, наиболее применимые, например, в обработке изображений. Однако вне зависимости от сложности у нейросетей есть общий недостаток: входные и выходные данные имеют фиксированный, заранее обозначенный объём, к примеру, изображение 300 на 300 пикселей или последовательность из 512 бит. С точки зрения математики нейросеть ведет себя как обыкновенная функция, хоть и сложно устроенная: у нее есть заранее обозначенное число аргументов и заранее заданный формат ответа. В качестве примера возьмём квадратичную функцию, принимающую один аргумент и выдающую одно значение. Данные особенности не являются препятствиями, если мы говорим о тех же изображениях или определенных последовательностях символов. Но как нейросеть справится с обработкой аудиодорожки или текста, строго говоря - любой условно бесконечной последовательности, в которой важно не только содержание, но и порядок следования информации? Для этих задач были придуманы рекуррентные нейросети. Их противоположности, носящие название «обычных», имеют более строгое название — нейросети прямого распространения (feed-forward neural networks) - в них информация передается только вперед по сети, от слоя к слою. В свою очередь в рекуррентных нейросетях нейроны обме-

ниваются информацией между собой: например, плюсом к новым входящим данным нейрон также получает некоторую информацию о прежнем состоянии сети. Таким образом в сети реализована «память», полностью меняющая характер ее работы и дающая возможность анализа любых последовательностей данных, в которых важен порядок следования значений — от звукозаписей до котировок акций.

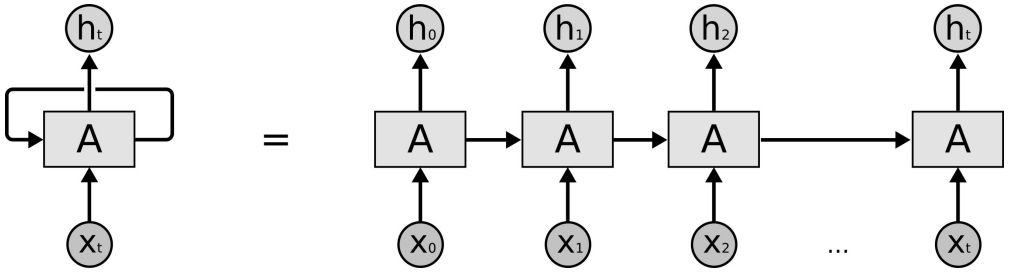


Рис. 2: Модель рекуррентной нейронной сети

Можно заметить, что RNN напоминают цепочку – этот факт говорит о тесной связи с последовательностями и списками. RNN – самая естественная архитектура нейронных сетей для работы с подобными данными.

В класс рекуррентных сетей с глобальной связью входят следующие основные архитектуры сетей:

1. Сети нелинейной авторегрессии с внешними входами, в них используется обратная связь между выходными и входными слоями;

2. Полносвязные рекуррентные сети с обратной связью между скрытым и входных слоем;

3. Рекуррентный многослойный перцептрон, содержащий более одного скрытого слоя, с обратными связями между каждым из расчета слоев и входным слоем;

4. Рекуррентные сети второго порядка, использующие нейроны второго порядка;

Первые три типа рекуррентных сетей допускают использование пространства состояний для изучения динамики среды. Этот подход берет начало в современной теории управления и является мощным методом изучения нелинейной динамики рекуррентных сетей. В обучении рекуррентных сетей можно выделить различные по своей сути подходы:

1. Использование известных соотношений между параметрами рекуррентной нейронной сети и ее динамикой (например, нейронная сеть Хакена и другие рекуррентные нейронные сети, в которых через задание весовых коэффициентов возможно закодировать желаемые фазовые портреты на правилах Хэбба);

2. Применение методов обучения с учителем на базе алгоритмов оптимизации по типу алгоритма обратного распространения ошибки (например, алгоритмы обратного распространения во времени, рекуррентное обучение в реальном времени, рекуррентное обратное распространение и алгоритмы, использующие Калмановскую фильтрацию);

3. Использование алгоритма обучения с учителем при рассмотрении сети в качестве не полной рекуррентности; данные с обратных связей можно рассмотреть как отдельные входные сигналы – контекстные нейроны, в итоге задача обучения упрощается и сводится к простым алгоритмам обучения нейронных сетей прямого распространения; к этому типу обучения можно отнести нейронные сети Элмана, Джордана и другие;

4. Использование алгоритмов обучения без учителя (инаптическая пластичность импульсных нейронных сетей, правило обучения Кохонена);

5. Отсутствие обучения в традиционном понимании изменения параметров системы (неявное обучение) – настройка весов случайным образом; при этом обучение выполняет специальное устройство – считыватель, производящий классификацию динамики сети; данный принцип является основой новой парадигмы нейронных сетей – резервуарных вычислений.

Главной идеей резервуарных вычислений является использование рекуррентной нейронной сети в качестве своеобразного резервуара с богатой динамикой и большими вычислительными возможностями. Этот резервуар наполняется случайным образом – это лишает необходимости проводить

его обучение. При подаче на вход резервуара непрерывного сигнала он начинает работать в одном из заранее заданных динамических режимов (состояний), зависящих от входного сигнала. Формирование резервуара происходит таким образом, что для похожих входных сигналов эти состояния были похожими, а для разных – разными. Выход резервуара ведёт на специальные устройства – считыватели, решающие поставленную в зависимости от состояния резервуара задачу – кластеризацию, классификацию, предсказание. Считывателями в данном случае могут выступать статические обучающие машины с простыми алгоритмами обучения. В итоге резервуар включает в своем состоянии динамику входного образа, а считыватели в зависимости от состояния резервуара распознают входной образ.

Постановка задачи

Целями настоящей работы является разработка нейросетевого метода обработки хаотических процессов, происходящих в динамике нелинейных систем.

Для реализации заявленных целей требуется:

1. Задать вид и способ построения нейронной сети, определяющей расхождение соседних траекторий.

2. Найти старший показатель Ляпунова, характеризующий наличие хаотических процессов.

3. Выяснить области параметров, отвечающих регулярным и хаотическим движениям.

4. Продемонстрировать работу нейросетевого метода определения старшего показателя Ляпунова для одного класса нелинейных систем.

Обзор литературы

Модель искусственного нейрона была предложена Уорреном МакКаллоком (Warren McCulloch) и Уолтером Питтсом (Walter Pitts) в 1943 году в работе [20]. В качестве основы для своей модели авторы использовали биологический нейрон. Искусственный нейрон МакКаллока—Питтса имеет N входных бинарных величин $x_1, ..., x_n$, которые трактуются как импульсы, поступающие на вход нейрону (рис. 3). В нейроне импульсы складываются с весами $w_1, ..., w_n$.

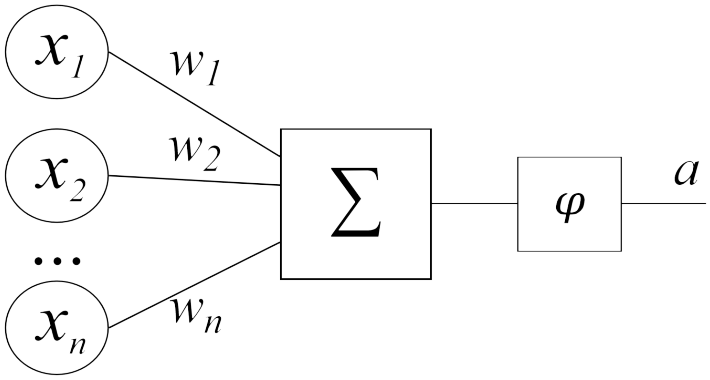


Рис. 3: Модель искусственного нейрона МакКаллока—Питтса

Авторами также был предложен метод объединения отдельных нейронов в искусственные нейронные сети. Для этого выходные сигналы нейрона передаются на вход следующему нейрону (рис. 4). Нейронная сеть содержит несколько слоев, на каждом из которых может находиться несколько нейронов. Слой, принимающий сигналы из внешнего мира, именуется входным. Слой, передающий сигналы во внешний мир, — выходным. Прочие слои называются скрытыми.

Заданные подобным образом искусственные нейронные сети могут приблизить любую непрерывную функцию с необходимой требуемой точностью [9–12]. Однако в настоящее время не существует конструктивного подхода, позволяющего гарантированно создавать нейронные сети с заранее заданными свойствами. Этот факт является серьёзным недостатком, ограничивающим применение нейронных сетей

Впервые идея обучения нейронных сетей была предложена Дональдом Хеббом (Donald Hebb) в 1949 году [13]. Согласно Хеббу, активирующие-

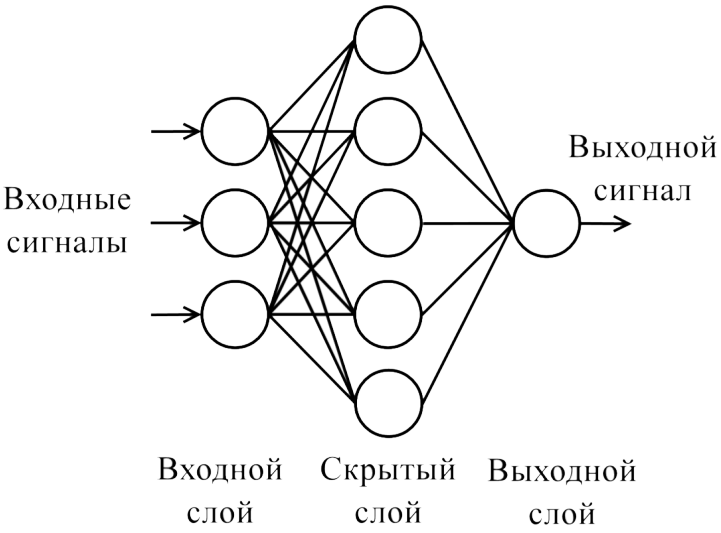


Рис. 4: Искусственная нейронная сеть

ся вместе связи нейронов должны усиливаться, а срабатывающие отдельно друг от друга – ослабевать.

Очень глубокий обучатель, предложенный в работе 1992 года [16], решает проблему исчезающего градиента и производит обучение сети глубиной до сотен слоев, используя предварительное обучение без учителя иерархии рекуррентных нейронных сетей. Каждая рекуррентная сеть обучается отдельно для предсказания следующего значения, поступающего ей на вход [2, 3].

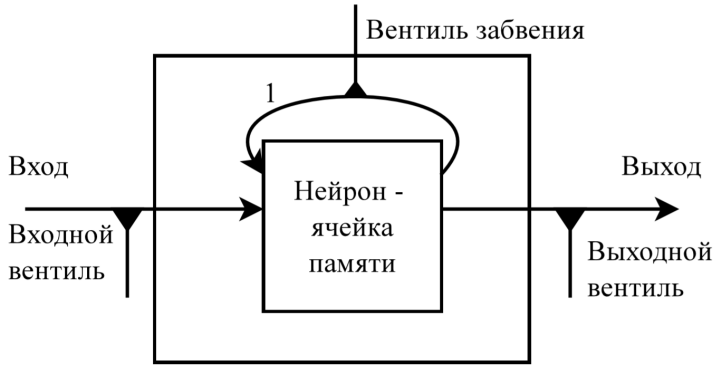


Рис. 5: Схема блока сети долго-краткосрочной памяти

Также проблему исчезающего градиента позволяет решить другой вариант рекуррентной нейронной сети — сеть долго–краткосрочной памяти (Long Short-Term Memory) [21–23]. Подобные сети содержат узлы специального типа, позволяющие хранить значения на протяжении долгого

времени. Блок сети долго-краткосрочной памяти содержит специальный нейрон, который используется в качестве ячейки памяти (рис. 5).

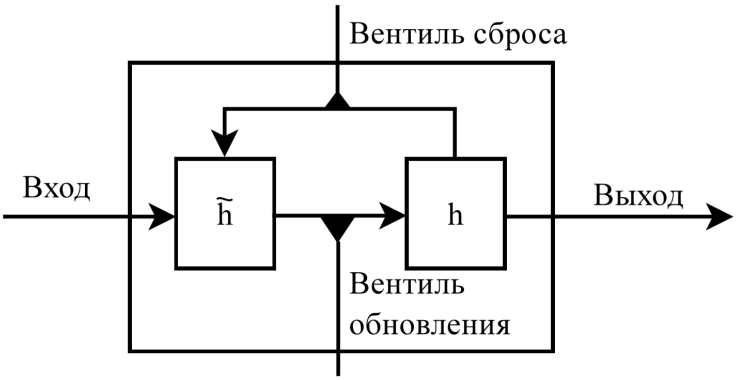


Рис. 6: Блок управляемой рекуррентной нейронной сети [18]

В 2014 году также была предложена новая структура рекуррентных нейронных сетей — управляемые рекуррентные нейронные сети (gated recurrent neural networks) [18, 19]. Данные сети похожи на сети долго-краткосрочной памяти, но в них не используются ячейки памяти. Схема блока управляемой рекуррентной сети показана на рис 6.

Глава 1. Хаотическая динамика нелинейных процессов

1.1. Динамический параметр хаотического поведения движений

Рассмотрим точку $x\,(t_0)$ которая является частью цикла динамической системы в начальный момент времени t_0. Выбрав маленькое число $\varepsilon\,(t_0)$, выберем ещё одну точку на цикле $\tilde{\mathbf{x}}\,(t_0)$ так, чтобы

$$\|\tilde{\mathbf{x}}\,(t_0) - \mathbf{x}\,(t_0)\| = \varepsilon\,(t_0)\,.$$

Через некоторый промежуток времени Δt точки $x\,(t_0)$ и $\tilde{\mathbf{x}}\,(t_0)$ преобразовываются пропорционально в $x(t)$ и $\tilde{\mathbf{x}}(t)$, при этом расстояние между ними будет $\varepsilon(t)$, где $t = t_0 + \Delta t$ (рис. 7).

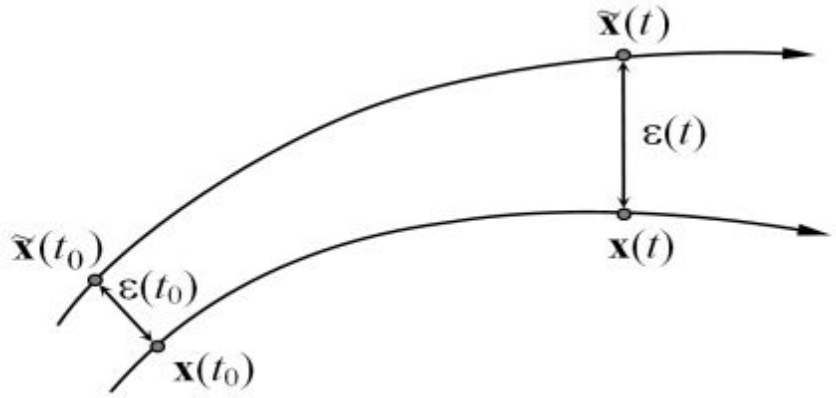


Рис. 7: Определению понятия экспоненты Ляпунова

Возможно такое, что $\varepsilon(t)$ зависящий от начального положения $x\,(t_0)$ и $\tilde{\mathbf{x}}\,(t_0)$, промежутка времени Δt и, разумеется, параметров данной динамической системы. Однако, в более кратком виде, можем представить, что

$$\varepsilon(t) \cong \varepsilon\,(t_0)\,e^{\lambda \Delta t}, \tag{1}$$

где λ — параметр, который называется *старший показатель Ляпунова*, определяемый рассматриваемой динамической системой.

Тогда, после простых преобразований, получим:

$$\lambda \cong \frac{1}{\Delta t} \ln \frac{\varepsilon(t)}{\varepsilon\left(t_0\right)} \tag{2}$$

Здесь стоит отметить, что в последнем соотношении, из-за того, что цикл ограничен (а значит ограниченности $\varepsilon(t)$) Δt может увеличиваться, пока $\varepsilon(t)$ много меньше размеров цикла, иначе $\lambda = 0$, при $\Delta t \to \infty$. А также, найденное в соответствии λ необходимо рассматривать как среднее по всем начальным точкам $x\left(t_0\right)$ цикла системы.

Учитывая всё выше сказанное, можем определить старший показатель Ляпунова как:

$$\lambda = \underset{\mathbf{x}(t_0) \in A}{E} \left\{ \begin{array}{c} \lim_{\Delta t \to \infty} \frac{1}{\Delta t} \ln \frac{\varepsilon(t)}{\varepsilon(t_0)} \\ \varepsilon(t) < \operatorname{diam} A \end{array} \right\}, \tag{3}$$

где A — цикл системы, $\operatorname{diam} A$ — его диаметр, а $E\{\cdot\}$ — математическое ожидание.

Старший показатель Ляпунова представляет уровень экспоненциального расхождения близлежащих траекторий. Присутствие у системы положительной экспоненты Ляпунова, свидетельствует об этом, что две любые траектории в течении некоторого времени быстро разойдутся, в таком случае есть зависимость от первоначальных условий. Вследствие этого, нахождение экспоненты Ляпунова предоставляет вероятность найти динамическую систему от наличия в ней хаотического действия.

Алгоритм оценки старшего показателя Ляпунова

Для оценки старшего показателя Ляпунова чаще всего используют алгоритм Бенеттина. Пусть имеется $\mathbf{x}_0$ — точка, являющаяся частью цикла А исследуемой динамической системы. Траекторию эволюции точки $\mathbf{x}_0$ назовём опорной траекторией. Зададим величину больше нуля ε удовлетворяющей условию $\varepsilon \ll \operatorname{diam} A$ будем выбирать такую точку возмущения $\tilde{\mathbf{x}}_0$ чтобы выполнялось равенство $\|\tilde{\mathbf{x}}_0 - \mathbf{x}_0\| = \varepsilon$ Рассмотрим эволюцию выбранных точек $\mathbf{x}_0$ и $\tilde{\mathbf{x}}_0$ в течении малого интервала времени T, найденные точки обозначим через $\mathbf{x}_1$ и $\tilde{\mathbf{x}}_1$ соответственно. Вектор $\Delta\mathbf{x}_1 = \tilde{\mathbf{x}}_1 - \mathbf{x}_1$ будем называть вектором возмущения, а его длину $\|\Delta\mathbf{x}_1\|$ амплитудой возмуще-

ния. Уже сейчас можно впервые оценить величину λ:

$$\tilde{\lambda}_1 = \frac{1}{T} \ln \frac{\|\Delta \mathbf{x}_1\|}{\varepsilon}. \tag{4}$$

Возьмем временной интервал T таким, чтоб значение нашей амплитуды возмущения осталось меньше линейных размеров неоднородностей фазового пространства и, конечно же , размеров цикла. Далее перейдем к рассмотрению перенормированного вектора возмущения.

$$\Delta \mathbf{x}_1' = \frac{\Delta \mathbf{x}_1}{\|\Delta \mathbf{x}_1\|} \varepsilon \tag{5}$$

и соответствующую ему новую точку возмущения $\tilde{\mathbf{x}}_1' = \mathbf{x}_1 + \Delta \mathbf{x}_1'$ Далее, продолжая данную процедуру, рассматривая вместо точек $\mathbf{x}_0$ и $\tilde{\mathbf{x}}_0$, точки $\mathbf{x}_1$ и $\tilde{\mathbf{x}}_1'$ соответственно (рис. 8).

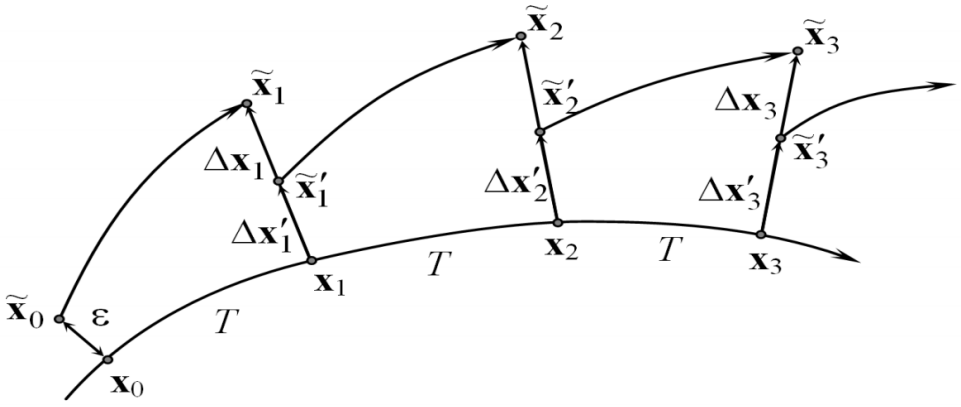


Рис. 8: К оценке экспоненты Ляпунова с помощью алгоритма Бенеттина

Проведя данную процедуру M раз, мы можем обозначить λ как среднее арифметическое величин $\tilde{\lambda}_i$ полученных на каждом шаге вычислений:

$$\lambda \cong \frac{1}{M} \sum_{i=1}^{M} \tilde{\lambda}_i = \frac{1}{M} \sum_{i=1}^{M} \frac{1}{T} \ln \frac{\|\Delta \mathbf{x}_i\|}{\varepsilon} = \frac{1}{MT} \sum_{i=1}^{M} \ln \frac{\|\Delta \mathbf{x}_i\|}{\varepsilon} \tag{6}$$

Логично, что для более точной оценки, необходимо выбрать значение M максимально возможным и производить вычисления для различных точек $\mathbf{x}_0$.

1.2. Анализ старшего показателя Ляпунова по наблюдениям

Когда уравнения динамической системы нам неизвестны, и, в общей картине, не представляется возможным определить все фазовые координаты ,и ведь мы часто сталкиваемся с такой ситуацией. Поэтому в таких ситуациях, для анализа старшего показателя Ляпунова можно использовать решение, которое было предложено Вольфом и основано на теореме Такенса и так же уже рассмотренном алгоритме Бенеттина. Этот метод строится на обработке измерений одной координаты рассматриваемой системы, что как раз нам сейчас и необходимо. Как уже было сказано, Такенс доказал, система, реконструируя методами временных задержек одной из фазовых координат, будет сохранять необходимые динамические и геометрические свойства, если будет выполнено неравенство:

$$m \geqslant 2[d] + 1, \tag{7}$$

где m — размерность пространства, d — фрактальная размерность, для текущего цикла, $[\cdot]$ — её сохранённая часть. В этом утверждении под фразой важнейшие динамические свойства понимается хаотичность и диссипативность системы, а под важными геометрическими свойствами — топологические инварианты такие, к примеру, как фрактальная размерность.

Выполнив неравенства (7) даёт возможность сохранить эти свойства, но эксперименты показывают, что данная нижняя оценка размерности пространства немного завышена. Некоторые хаотические системы сохраняют выше показанные свойства даже при $m = [d + 1]$.

Допустим, у нас есть временной ряд $x(t)$, $t = 1, \ldots, N$ измерений для одной координаты текущего хаотического процесса, показанные через одинаковые моменты времени. В этом случае, используя имеющиеся методы, нужно найти размерность пространства m и задержку T. При помощи преобразований, можем получить набор точек пространства $\mathbf{R}^m$:

$$\mathbf{x}_i = [x(i), x(i - \tau), \ldots, x(i - (m - 1) \cdot \tau)], \tag{8}$$

где $i = [(m - 1)\tau + 1], \ldots, N$.

Из последовательности (8) возьмём точку и выделим ее через $\mathbf{x}_0$. Анализируя последовательность (8), будем брать такую точку $\tilde{\mathbf{x}}_0$ нужную для выполнения соотношения $\|\tilde{\mathbf{x}}_0 - \mathbf{x}_0\| = \varepsilon_0 < \varepsilon$,, где ε - фиксированная величина, значительно меньшая размеров восстановленного фозового портрета. При этом нужно, точки $\mathbf{x}_0$ и $\tilde{\mathbf{x}}_0$ были разделёнными по времени. После этого рассматриваем эволюцию подобранных точек на восстановленном фазовом цикле до того момента, пока величина не станет превышать расстояние между ними ε_{max}. Обозначим полученные точки через $\mathbf{x}_1$ и $\tilde{\mathbf{x}}_1$, а расстояние - ε_0', интервал времени эволюции — T_1.

Затем, снова обращая внимание на последовательность (8), , нужно найти такую точку $\tilde{\mathbf{x}}_1'$, чтобы она была максимально близка к $\mathbf{x}_1$ т.е. $\|\tilde{\mathbf{x}}_1' - \mathbf{x}_1\| = \varepsilon_1 < \varepsilon$, а векторы $\tilde{\mathbf{x}}_1 - \mathbf{x}_1$ и $\tilde{\mathbf{x}}_1' - \mathbf{x}_1$ имели, если это возможно, одинаковое направление. Затем эти действия повторяются, но вместо $\mathbf{x}_0$ и $\tilde{\mathbf{x}}_0$ нужно рассмотреть точки $\mathbf{x}_1$ и $\tilde{\mathbf{x}}_1'$ (рис. 9). После повторения данной

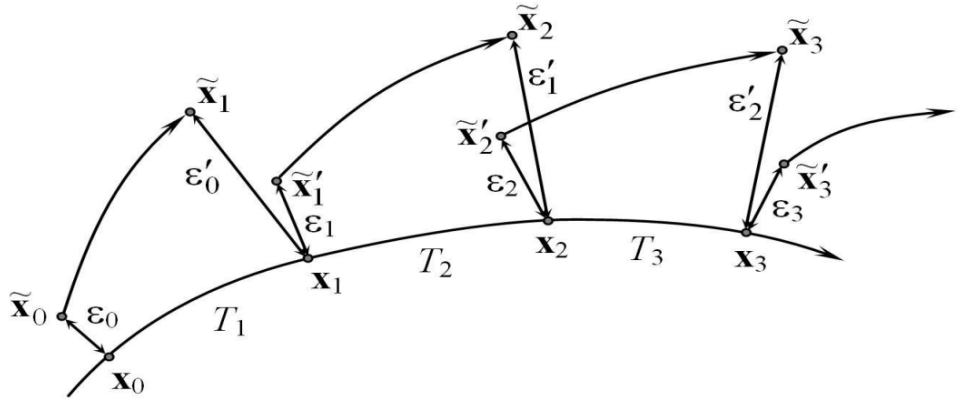


Рис. 9: Анализ экспоненты Ляпунова по сведениям одной из наблюдаемых координат..

процедуры M количество раз, можем оценить экспоненту Ляпунова так:

$$\lambda \cong \frac{\sum_{k=0}^{M-1} \ln\left(\varepsilon_k'/\varepsilon_k\right)}{\sum_{k=1}^{M} T_k}. \tag{9}$$

Приведённый метод основывается на теореме [16], которая утверждает, что расхождение двух случайно выбранных точек на цикле с малым шансом описывает старший показатель Ляпунова.

Глава 2. Исследование динамики нелинейных систем

2.1. Математическая модель одной нелинейной системы

Рассмотрим модель одной нелинейной системы, заданной в полярной системе координат (R, θ, z) и определяемой в 3-х мерном пространстве дифференциальными уравнениями

$$\ddot{\mathbf{r}} = \nabla\varphi, \tag{10}$$

где $\mathbf{r}$ — радиус-вектор движения материальной точки, φ — вещественная функция координат (R, θ, z), оператор ∇ имеет вид:

$$\nabla = \left(\frac{\partial}{\partial R}, \frac{1}{R}\frac{\partial}{\partial \theta}, \frac{\partial}{\partial z}\right)^T$$

и R — радиус, θ — угол вращения, z — вертикальная ось.

Движение материальной точки полностью определяется решением задачи Коши с соответствующими этому решению начальными данными:

$$\mathbf{r}(t_0) = (R_0, \theta_0, z_0)^T \quad \text{и} \quad \mathbf{v}(t_0) = (v_{R0}, v_{\theta 0}, v_{z0})^T,$$

где $\mathbf{v} = \dot{\mathbf{r}}$ — новая переменная, задающая вектор скорости движения этой точки. Саму функцию φ возьмём определённого вида, формульная запись которого есть:

$$\varphi(\xi) = \frac{\alpha}{\alpha - 1 + w(\xi)},$$

$$w(\xi) = \left(1 + \alpha^p \xi^p\right)^{1/p},$$

при важном предположении, что параметр ξ удовлетворяет равенству

$$\xi^2 = R^2 + z^2 + 2\big(1 - \varepsilon\big)\left(\sqrt{z^2 + \varepsilon^2} - \varepsilon\right),$$

в случае принятых параметров $\alpha > 0, p > 0, \varepsilon \in [0, 1]$.

Для данной системы дифференциальных уравнений существуют со-

ответствующие интегралы E и I:

$$E = \frac{1}{2}\Big(v_R^2 + v_\theta^2 + v_z^2\Big) - \varphi(\xi), \quad I = Rv_\theta,$$

где v_R, v_θ, v_z — элементы скорости в системе цилиндрических координат R, θ, z. Так как, в данных предположениях, в принятой системе функция φ является независящей переменной θ, то начальная система (10) окончательно запишется как:

$$\begin{aligned} \dot{R} &= v_R, \qquad & \dot{v}_R &= \frac{I^2}{R^3} + \frac{\partial \varphi}{\partial R}, \\ \dot{\theta} &= \frac{I}{R^2}, \qquad & \dot{v}_z &= \frac{\partial \varphi}{\partial z}, \\ \dot{z} &= v_z. \end{aligned} \tag{11}$$

В последующем будем моделировать движение материальной точки при допущении $z = 0$ (а также начальных значениях $z_0 = 0$ и $v_{z0} = 0$). Поэтому система (11) может быть упрощена до вида всего лишь трех уравнений: и, с учётом вида функции φ, окончательно получаем систему

$$\begin{aligned} \dot{R} &= v_R, \\ \dot{\theta} &= \frac{I}{R^2}, \\ \dot{v}_R &= \frac{I^2}{R^3} - \frac{\alpha^{p+1} R^{p-1}\Big(1 + \alpha^p R^p\Big)^{(1-p)/p}}{\Big(\alpha - 1 + \big(1 + \alpha^p R^p\big)^{1/p}\Big)^2}. \end{aligned} \tag{12}$$

Таким образом, рассмотрели нелинейную систему дифференциальных уравнений, точной динамика которой не может быть получена непосредственным её интегрированием. Поэтому требуется численной решение данной системы и определение её основных динамических характеристик по результатам численных решений, принятых в качестве наблюдательных данных. Анализ этой информации будем производить всеми возможными способами, в том числе и нейросетевой обработкой временных рядов "наблюдательных"данных.

2.2. Восстановление фазовой динамики по наблюдательным данным

Рассмотрим теперь задачу восстановления по экспериментальным данным, полученным в результате численного решения исследуемой нелинейной системы дифференциальных уравнений, какой-либо качественной информации о динамике изучаемых процессов. Такой качественной информацией будет поведение движений в фазовом (или его срезе) пространстве представленной системы. Как правило, в фазовом пространстве можно определить наличие странных аттракторов, предельных циклов, периодических траекторий и т.п.

Для ряда наборов модельных параметров произведём численной интегрирование и, по полученным экспериментальным данным, на основе нейросетевого метода обработки данных временных рядов определим наличие возможных аттракторов в фазовом пространстве.

Возьмём сначала следующие значения модельных параметров:

$$I = 0.7016, \quad R = 1.2,\ \theta = 0,\ v_R = 0,$$

а также вполне определённые величины параметров

$$p = 0.4 \quad \text{и} \quad \alpha = 2.$$

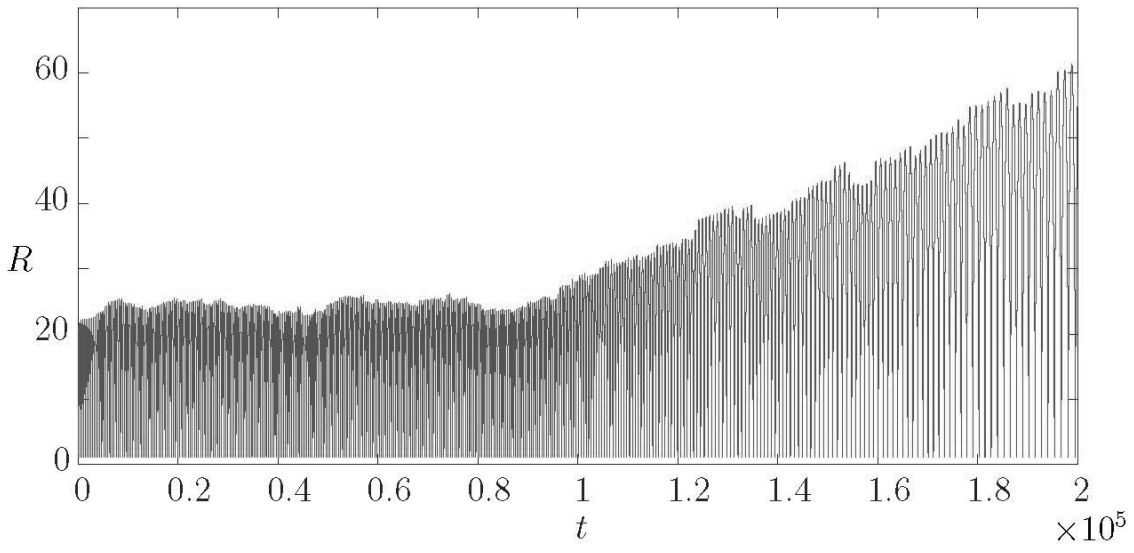


Рис. 10: Наблюдательные данные по переменной R

Сгенерируем экспериментальные данные наблюдений по переменной R и отобразим их на соответствующем графике (Рис. 10). Будем рассматривать эти данные как если бы математическая модель динамики процесса была бы неизвестной, а эти данные являлись результатом практических наблюдений. Тогда получим временной ряд, который будет рассматриваться в дальнейшем как данные экспериментальных наблюдений и по этим данным будем восстанавливать фазовый портрет системы (12):

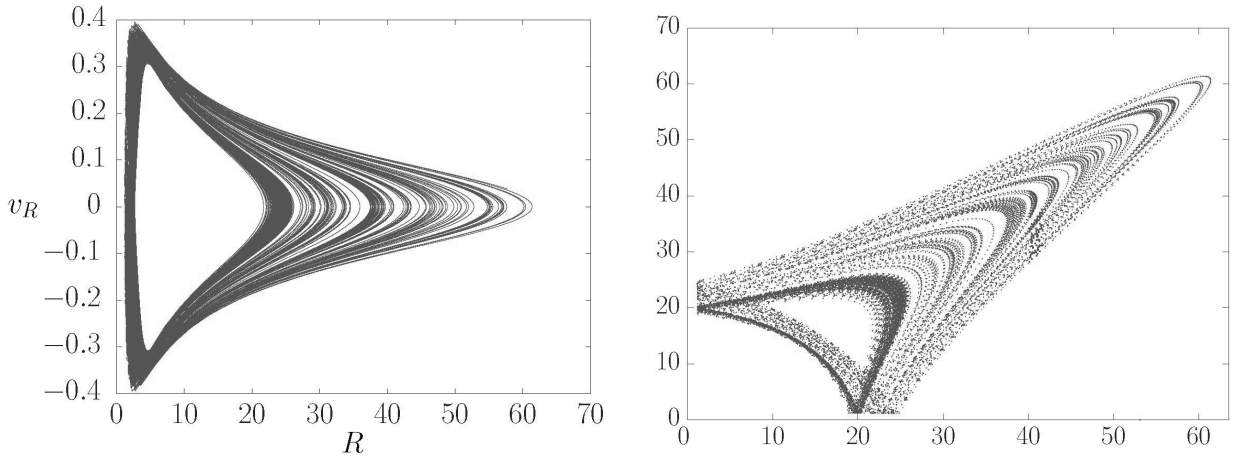
$$R_i = R(t_i), \quad t_i = 10i, \ i = 0, 1, \ldots, 20000$$

отображены в табл. 1.

Таблица 1: Наблюдательные данные

t_i	0	10	20	...	199980	199990	200000
R_i	1.200000	4.263244	7.127422	...	56.975403	57.380087	57.766636

Графики реального фазового портрета системы (12) и восстановленного, с помощью нейросети показаны на рис. 11.



а) б)

Рис. 11: а) Фазовый портрет, б) Восстановленный портрет

Таким образом восстановленный фазовый портрет хорошо отражает фактическую динамику процесса движений системы (12) при выбранных параметрах.

Далее, возьмём теперь другой набор параметров системы:

$$I = 0.4, \quad R = 0.5, \ \theta = 0 \quad \text{и} \quad v_R = 0.5,$$

в этот раз при величинах

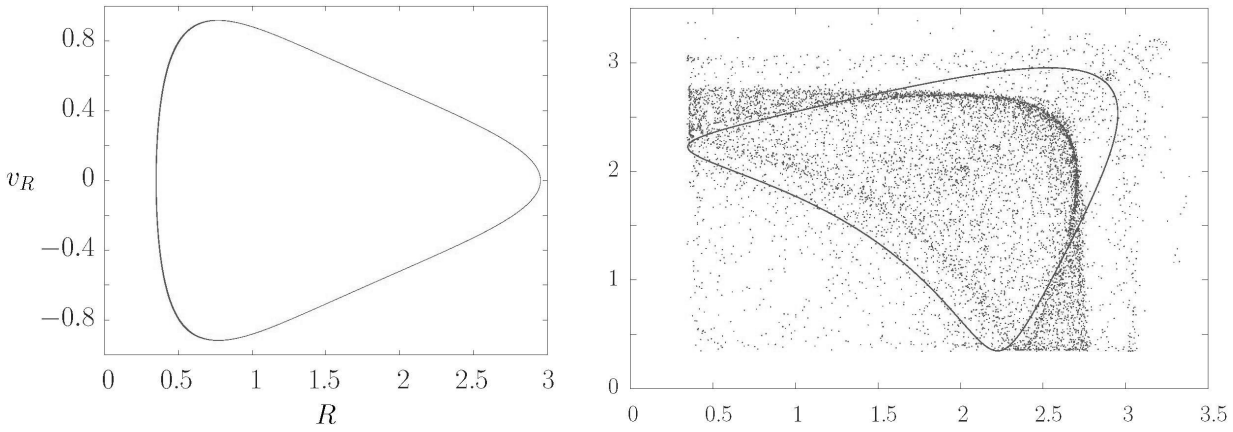
$$p = 3 \quad \text{и} \quad \alpha = 1.$$

Снова построим соответствующий график, на котором указано изменение по времени радиуса R а в табл. 2 временной ряд, используемый как наблюдательные данные $R_i = R(t_i)$, при $t_i = 10i$ и $i = 0, 1, \ldots, 20000$.

Таблица 2: Экспериментальные данные по переменной R

t_i	0	10	20	...	199980	199990	200000
R_i	0.500000	1.736379	0.867515	...	2.726192	1.492619	1.386241

И снова восстановим возможный вид картинки фазового пространства (Рис. 12).



а) б)

Рис. 12: а) Фазовый портрет, б) Восстановленный портрет

Результат восстановленного поведения на фазовом пространстве полностью отражает факт установившегося периодического движения динамической системы.

На рис. 12 а) нарисован график последних циклов движения решения системы при выбранных параметрах, и этот график указывает на существование периодического движения материальной точки. На рис. 12 б) показан процесс восстановления фазового портрета нейронной сетью, который также подтверждает сначала наличие произвольных колебаний движения, а затем постепенной их затухание к переходу на периодическое движение.

Глава 3. Нейросетевое вычисление параметров хаотической динамики

3.1. Нейросетевой анализ старшего показателя Ляпунова по наблюдениям

Представленный ранее метод имеет большую вычислительную сложность и невозможность применения для небольшого объема исходных данных. Это происходит по причине трудности нахождения двух точек из ряда, находящихся друг от друга на расстояние меньше, чем 10^{-8}. В особенности это проблемно для настоящих сведений. Можно с этим справиться, используя нейронные сети для нахождения старшего показателя Ляпунова.

Основная идея данного метода — вычисление при помощи нейросетей, расходящихся рядом траекторий, расположенных на N шагов вперёд. Нейросеть состоит из $k \geqslant m-1$ входных, скрытых - p и единственного выходного нейрона (рис. 13). В данном случае m — размерность пространства вложения.

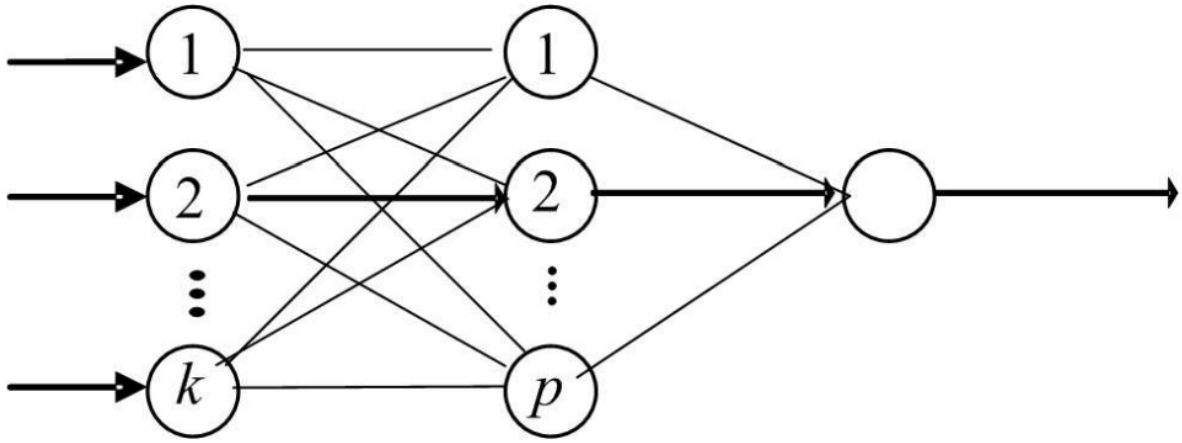


Рис. 13: Модель нейронной сети.

Изначально нужно обучить нейронную сеть методу скользящего окна, чтобы она могла прогнозировать:

$$x(t+i\tau) = F[x(t+(i-1)\tau), x(t+(i-2)\tau), \ldots, x(t+(i-k)\tau)], \quad (13)$$

для всех $i = 1, \ldots, N$.

После обучения нейросети, можно осуществить движение рядом стоящих точек на фазовой траектории, применяя итерационный метод. Данная процедура может быть представлена следующим алгоритмом:

- Проводим обучение нейросети по методу скользящего окна.
- Берём какую-нибудь точку $x(t)$ из обучающей выборки и создаёт новую выборку данных: $x(t), x(t-\tau), \ldots, x(t-(k-1))\tau$, где k - размер окна.
- Применяя многошаговый прогноз, высчитываем $x(t+\tau), x(t+2\tau), \ldots,$ $\ldots, x(t+n\tau)$, $x(t+i\tau) = F[x(t+(i-1)\tau), x(t+(i-2)\tau), \ldots, x(t+(i-k)\tau)]$, при $i = \overline{1,n}$, F — нелинейная функция.
- $x'(t) = x(t) + d_0$, где $d_0 \approx 10^{-8}$, предоставляя нейросети $x'(t), x(t-\tau), \ldots, x(t-(k-1)\tau)$, повторяем шаг 3 для нахождения $x'(t+i\tau), i = \overline{1,n}$.
- Даём оценку $\ln d_i = \ln |x'(t+i\tau) - x(t+i\tau)|, i = \overline{1,n}$ можем выбрать только такие точки, при которых $\ln d < 0$.
- Выполняем построение графика $\ln(d_n)$ от n.
- Нужно построить прямую регрессии для данных точек и определяем наклон, равный максимальному показателю Ляпунова.

Этот метод вычисления λ был испытан на многослойном персептроне. Была использована нейронная сеть с 7 входными, 5 скрытым и 1 выходным нейронным элементом для прогнозирования временных рядов. Скрытые слои вычислены при помощи сигмоидной функции, а выходной нейрон — линейной. Обучение нейросети происходило при помощи метода обратного распространения ошибки. В результате для приведённых выше случаев параметров показано, что нейронная сеть дает довольно точную оценку показателя Ляпунова даже и при небольшом объеме исходных данных. Очевидное преимущество нейросетевого подхода в сравнении с традиционным — точность и простота.

3.2. Определение параметров динамики для регулярных и не регулярных типов движений

Возьмём теперь рассмотренные выше наборы параметров и определим значения старших показателей Ляпунова для исследуемой динамической системы.

Сначала примем:

$$I = 0.7016, \quad R = 1.2, \ \theta = 0, v_R = 0$$

и

$$p = 0.4 \quad \text{и} \quad \alpha = 2.$$

Тогда график вычисления старшего показателя Ляпунова в этом случае будет иметь вид:

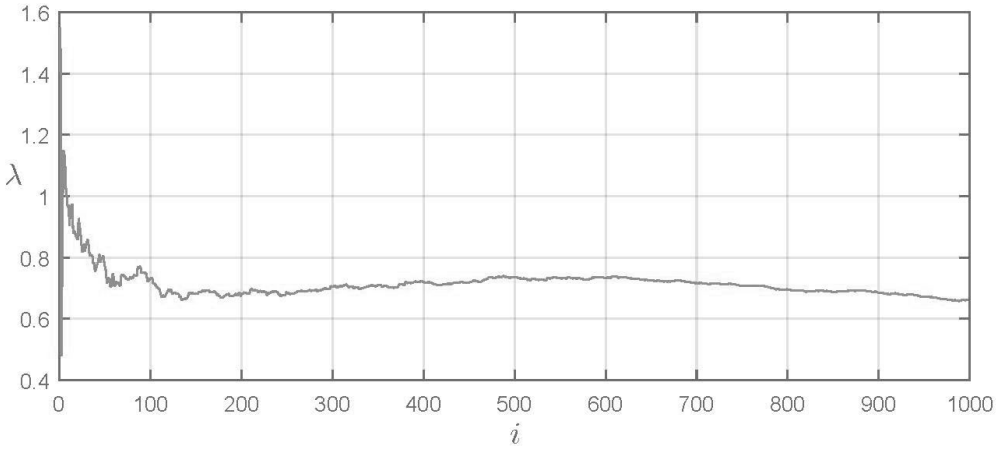


Рис. 14: Расчёт старшего показателя Ляпунова, $\lambda = 0.6614$

Положительное значение старшего показателя Ляпунова отражает отсутствие периодического движения. Движение близких точек на соседних траекториях быстро приводит к их обоюдному разбеганию.

Рассматривая второй случай параметров

$$I = 0.4, \quad R = 0.5, \ \theta = 0 \quad \text{и} \quad v_R = 0.5,$$

в этот раз при величинах

$$p = 3 \quad \text{и} \quad \alpha = 1$$

вычисляем следующее значение старшего показателя Ляпунова:

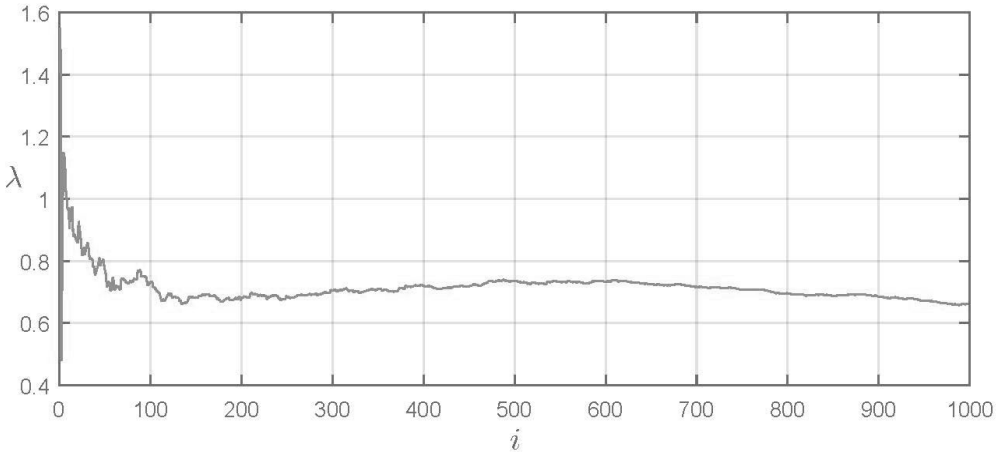


Рис. 15: График старшего показателя Ляпунова, $\lambda \leq 0$

После 8000 дискретных отсчётов времени расчётные данные наблюдений замыкаются в периодический временной ряд. Близкие точки на соседних траекториях больше не разбегаются, их близость остаётся неизменной. И сам старший показатель Ляпунова является не положительным, что характеризует периодическое движение.

Выводы

В данной работе рассматривается подход расчёта динамических характеристик существенно нелинейных динамических процессов. Часто для таких процессов математические модели или не определены или задаются недостаточно точно, но сами данные наблюдения динамики этих явлений могут быть доступны в виде временных рядов. Показан метод построения нейронной сети, которая может по данным наблюдения рассчитывать важные динамические характеристики, в том числе такие как старший показатель Ляпунова — мера наличия хаотических режимов поведения.

Заключение

Главной целью данной работы было разработка и реализация нейросетевого метода вычисления динамических параметров (старший показатель Ляпунова) на базе рассмотренной модели нелинейной динамики с хаотическими режимами.

Основные результаты представленной дипломной работы:

- Предложен метод создания нейронной сети для работы с временными рядами наблюдательных данных;
- Рассмотрена класс существенно нелинейных динамических систем с возможными как регулярными, так и хаотическими типами траекторий;
- Рассчитан старший показатель Ляпунова в локальной области фазового пространства;
- Определены возможные набора параметров указанной системы, отвечающих регулярным и хаотическим типам движений.

Список литературы

[1] Вапник В. Н., Червоненкис А. Я. Теория распознавания образов. — М.: Наука, 1974.

[2] Connor J., Martin D.R., Atlas L.E. Recurrent Neural Networks and Robust Time Series Prediction // IEEE Transactions on Neural Networks. 1994. Vol. 5, No. 2. P. 240–254.

[3] Dorffner G. Neural Networks for Time Series Processing // Neural Network World. 1996. Vol. 6. P. 447–468.

[4] Минский М., Пейперт С. Персептроны. — М.: Мир, 1971.

[5] Sompolinsky H., Crisanti A., H.J. Sommers Chaos in Random Neural Networks // Physical Review Letters. 1988. Vol.61, №3, 259-262

[6] Горбань А.Н., Дунин-Барковский В.Л., Кирдин А.Н. и др. Нейроинформатика. Новосибирск: Наука. 1998. С. 296.

[7] Бендерская Е. Н., Толстов А. А. Реализация осцилляторной хаотической нейронной сети с применением технологии NVIDIA CUDA для решения задач кластеризации // Информационно-управляющие системы. 2014. №4. С. 94–101.

[8] Галушкин А. И. Нейронные сети: основы теории. – М. : Горячая линия Телеком. 2010.

[9] Stone M.N. The Generalized Weierstrass Approximation Theorem. Mathematics Magazine. 1948. vol. 21, no. 4. pp. 167–184. DOI: 10.2307/3029750.

[10] Горбань А.Н., Дунин-Барковский В.Л., Кирдин А.Н. и др. Нейроинформатика. Новосибирск: Наука. 1998. С. 296.

[11] Hornik K., Stinchcombe M., White H. Multilayer Feedforward Networks are Universal Approximators. Neural Networks. 1989. vol. 2, no. 5. pp. 359–366. DOI: 10.1016/0893-6080(89)90020-8.

[12] Mhaskar H.N., Micchelli Ch.A. Approximation by Superposition of Sigmoidal and Radial Basis Functions. Advances in Applied Mathematics. 1992. vol. 13, no. 13. pp. 350–373. DOI: 10.1016/0196-8858(92)90016-P.

[13] Hebb D.O. The Organization of Behavior. New York: Wiley. 1949. 335 p

[14] Воронцов К. В. Обзор современных исследований по проблеме качества обучения алгоритмов // Таврический вестник информатики и математики. 2004. №1. С. 5–24.

[15] Измакова О. А. Рандомизированные алгоритмы самообучения для настройки ассоциативных нейронных сетей // Стохастическая оптимизация в информатике. 2005. Т. 1. №1-1. С. 81–102.

[16] Schmidhuber J. Learning Complex, Extended Sequences Using the Principle of History Compression // Neural Computation. 1992. Vol. 4, No. 2. P. 234–242.

[17] Grossberg S. Some Networks That Can Learn, Remember, and Reproduce any Number of Complicated Space-Time Patterns // International Journal of Mathematics and Mechanics. 1969. Vol. 19. P. 53–91. DOI: 10.1512/iumj.1970.19.19007.

[18] Cho K., van Merrienboer B., Gulcehre C., et al. Learning Phrase Representations Using RNN Encoder-Decoder for Statistical Machine Translation // Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP) (Doha, Qatar, October 25–29, 2014), 2014. P. 1724–1734. DOI: 10.3115/v1/d14-1179.

[19] Cho K., van Merrienboer B., Bahdanau D., et al. On the Properties of Neural Machine Translation: Encoder–Decoder Approaches // Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation (Doha, Qatar, October 25, 2014),

[20] McCulloch W.S., Pitts W. A Logical Calculus of the Ideas Immanent in Nervous Activity // The Bulletin of Mathematical Biophysics. 1943. Vol. 5, No. 4. P. 115–133. DOI: 10.1007/BF02478259.

[21] Hochreiter S., Schmidhuber J. Long Short-Term Memory // Neural Computation. 1997. Vol. 9, No. 8. P. 1735–1780. DOI: 10.1162/neco.1997.9.8.1735.

[22] Gers F.A., Schmidhuber J., Cummins F. Learning to Forget: Continual Prediction with LSTM // Neural Computation. 2000. Vol. 12, No. 10. P. 2451–2471.

[23] Perez-Ortiz J.A., Gers F.A., Eck D., et al. Kalman Filters Improve LSTM Network Performance in Problems Unsolvable by Traditional Recurrent Nets // Neural Networks.

[24] Granichin O., Volkovich V., Toledano-Kitai D. Randomized Algorithms in Automatic Control and Data Mining. – Berlin, Heidelberg:Springer. 2015.

[25] Rosenblatt F. Perceptron simulation experiments // Investigative Reporters and Editors Conference. 1960. Vol. 48. №3. P. 301–309.

[26] Minsky M., Papert S. Perceptrons. – Massachusetts:MIT Press. 1969.

[27] Цыпкин Я. З. Основы теории обучающихся систем. – М.: Наука. 1970.

Приложение

```
import numpy as np
import matplotlib.pyplot as plt
import time


# Network parameters

N = 200                            # Number of neurons
g_list = np.linspace(0.5,1.5,11) # List of nonlinearity gains

output_length = len(g_list)

# Simulation parameters

N_burn = 500     # Number of burn-in steps
N_sample = 1000  # Number of sample collection steps
dt = 0.1         # Time step size (dt = 1 & Euler Forward => discrete)
d0 = 1e-8        # Separation between nearby trajectories (for maximal

N_total = N_burn + N_sample      # Total number of steps to simulate.

# Initial activity distribution parameters

mean = 0
sigma = 1


def RNN(x, W, g):
  return -x + np.dot(W,np.tanh(g*x))

def RK4(x, W, g, dt, f):
  k1 = dt * f(x, W, g)
```

```
  k2 = dt * f(x + 0.5*k1, W, g)
  k3 = dt * f(x + 0.5*k2, W, g)
  k4 = dt * f(x + k3, W, g)
  return x + (1/6) * (k1 + 2*k2 + 2*k3 + k4)


lambda_list = np.zeros(output_length) # Maximal Lyapunov exponents lis

t_start = time.time()
t1 = time.time()

# Loop over g's
for i in range(len(g_list)):
  g = g_list[i]
  # Generate random connectivity matrix
  W = np.random.normal(0,1/np.sqrt(N),[N,N])
  # Initial state
  h1 = np.random.normal(mean,sigma,[N,1])          # Reference traject
  #print(h1)
  rand_dir = np.random.randn(N).reshape(h1.shape)
  h2 = h1 + d0 * rand_dir/np.linalg.norm(rand_dir) # Nearby trajectory

  lambda_ = 0 # Maximal Lyapunov exponent

 # Simulate

  for j in range(N_total):



    # Update trajectories

    h1 = RK4(h1, W, g, dt, RNN)
```

```
    h2 = RK4(h2, W, g, dt, RNN)


    # Calculate distance
    d1 = np.linalg.norm(h1-h2)


    # Re-align nearby trajectory
    h2 = h1 + (d0/d1)*(h2-h1)


    if j > N_burn:
      lambda_ = lambda_ + np.log(d1/d0) # Update Lyapunov exponent


  lambda_list[i] = lambda_/(dt*N_sample)


  t2 = time.time()
  print('g = ' + str(g) + ' runtime: ' + str(t2-t1) + ' s')
  t1 = t2
print('Total runtime: ' + str(time.time()-t_start) + ' s')
```

```
Plot

plt.figure()

p1 = plt.plot(g_list,lambda_list,'-ob',label='Simulation')

p2 = plt.plot(g_list,np.zeros(len(g_list)),'r') # 0

p3 = plt.plot(g_list, (g_list<1)*(-1+g_list) + (g_list>=1)*0.5*(g_list

plt.legend()

plt.xlabel('g');

plt.ylabel(r'lambda');

plt.title('Maximal lambda: N = ' + str(N));
```