

Нелинейная динамика обработки информации в нейронных сетях

А.Б. Потапов и М.К. Али

Аннотация

Рассматривается возможная роль сложной динамики и хаоса в работе нейронных сетей, обрабатывающих информацию. В начале дан обзор принципов работы некоторых наиболее известных типов нейронных сетей, а затем обсуждается ряд попыток использовать хаос в нейронных сетях. Основная цель работы — представить новый взгляд на проблему хаоса в задачах обработки информации. Показано, что хаос естественно возникает в задачах управления, где нейронная сеть является управляющей подсистемой более сложной системы. Показано, что такая сеть может использовать хаос в своей работе для освоения новых действий в методе обучения, называемом обучением поощрением (Reinforcement Learning). Обсуждаются также гамильтоновы нейронные сети.

1 Введение

Искусственные нейронные сети широко используются для решения как инженерных, так и научных задач [17, 18, 45, 46, 59, 58, 79, 95, 97]. Поскольку они оказались весьма эффективным инструментом обработки информации, постоянно делаются попытки расширить область их приложения или найти новые принципы их построения или работы. Около десяти лет назад предпринимались усилия с целью улучшить работу нейросетей при помощи динамического хаоса, обзор большинства таких попыток можно найти, например, в [93, 34], см. также [5]. Несмотря на множество интересных деталей, главный вопрос так и остался невыясненным: полезен или нет хаос для целей обработки информации? Аргументы существуют как за, так и против. С одной стороны, активность мозга весьма сложна и, по некоторым данным, хаотична. Быть может, эта хаотичность даёт какие-либо преимущества в борьбе за существование? С другой стороны, никто не делает хаотическими, скажем, процессоры для того, чтобы повысить их надежность или производительность. В таком случае, почему хаос будет полезен в нейронных сетях, которые также занимаются обработкой информации?

Цель данной статьи — рассмотреть принципы работы основных типов нейронных сетей с точки зрения нелинейной динамики, обсудить ряд попыток сделать динамику сети хаотической, а также представить альтернативный взгляд на роль хаоса в нейросетях. Если рассмотреть нейронную сеть как управляющий элемент некоторой большей системы, взаимодействующей с окружающим миром, то хаос может возникнуть уже в очень

простых моделях. Несколько таких моделей известны, например, в исследованиях по искусственному интеллекту. Более того, системам, взаимодействующим со своим окружением, необходим источник "инициативы" для того, чтобы исследовать мир и учиться на собственном опыте. И динамический хаос вполне может служить источником такой инициативы. Кроме того, настоящей работе мы обсудим гамильтоновы нейронные сети, которые практически не исследовались до настоящего времени. Такие сети могут оказаться классическими аналогами квантовых нейронных сетей.

Структура статьи следующая. Сначала в разделе 2 мы обсудим определение и различные взгляды на нейронные сети. В разделе 3 мы рассмотрим принципы работы некоторых существующих нейронных сетей для того, чтобы показать, почему, с точки зрения их пользователей и разработчиков сложная динамика и хаос не являются необходимостью. В разделе 4 мы поясним, почему хаос *может быть* полезным. В разделах 5, 6 и 7 мы обсудим альтернативный подход к проблеме.

2 Что такое искусственная нейронная сеть?

Несмотря на огромное число публикаций, в настоящее время нет общепринятого определения искусственной нейронной сети (ИНС). Причина, скорее всего, в том, что различных типов сетей существует очень много. Поэтому иногда используется более общий термин, "коннекционизм", обозначающий методологию создания сложной системы как комбинации похожих или идентичных элементов [29]. Для ИНС базовые нелинейные элементы называются формальными нейронами. Как правило, формальный нейрон получает несколько входных сигналов x_i , суммирует их, выполняет нелинейное преобразование над этой суммой $y = f(\sum x_i)$, после чего посылает y в качестве своего выходного сигнала другим нейронам или устройствам. Функция f называют функцией активации. Очень часто f является "сигмоидной" функцией вида $f(x) = 1/(1 + e^{-ax})$ или $f(x) = \tanh ax$. В предельном случае $a \rightarrow \infty$, f становится пороговой функцией.

Связи между нейронами характеризуются весами w_{ij} . Нейроны могут также получать внешние сигналы X_i с весами w'_{ik} . Таким образом, типичная операция, выполняемая нейроном в сети — это преобразование вида $f(\sum_j w_{ij}x_j + \sum_k w'_{ik}X_k)$. Выходом нейросети служат состояния нейронов (всех или некоторых) по окончании всех промежуточных расчетов. Таким образом, основные черты нейронной сети следующие:

- Это устройство для обработки информации, получающее на входе сигнал X (как правило, вектор) и вырабатывающее выходной сигнал $Y = F(X)$ (Y также может быть вектором).
- Она состоит из элементов-нейронов, которые работают параллельно, подобно нейронам мозга.
- Она способна обучаться. Обычно под обучением понимается процесс подбора весов w_{ij} для того, чтобы добиться желаемых свойств отображения F .

Искусственные нейронные сети (ИНС) исследуются с нескольких точек зрения, из которых наиболее важными нам представляются следующие четыре.

2.1 Нейросети в статистике и численных методах

На практике очень часто возникает задача аппроксимации неизвестной функции по экспериментальным точкам: задан набор пар точек $\{X_i, Y_i = \Phi(X_i)\}$, необходимо построить аппроксимацию $F(X)$ для $\Phi(X)$. Помимо обычной аппроксимации кривых и поверхностей, к этому виду можно привести поразительно большое количество задач распознавания образов, классификации, прогноза и управления. Например, в качестве X_i могут выступать показания датчиков, характеризующих обстановку вокруг автомобиля, а в качестве Y_i — действия опытного водителя в данной ситуации. Хорошая аппроксимация $F(X)$ позволяет создать систему, способную управлять автомобилем.

Классическая теория аппроксимации, как правило, имеет дело с функциями одной переменной. При необходимости построить аппроксимацию, скажем, функции 10 переменных классические методы, такие как аппроксимация полиномом, не слишком эффективны. Аппроксимации комбинациями сигмоидных функций часто оказываются существенно лучше.

Общая задача аппроксимации нейронной сетью выглядит следующим образом. Выбирается некоторый общий вид аппроксимирующей функции $F(X, w)$ ("архитектура сети"), а затем подбираются коэффициенты w с тем чтобы минимизировать погрешность $\sum |Y_i - F(X, w)|^2$. Процесс минимизации называется обучением нейросети. Он существенно зависит от выбранного типа сети.

На первый взгляд все выглядит исключительно просто. Но если начать строить реальную сеть, немедленно возникают вопросы: сколько нейронов взять, как их соединять, какие методы минимизации использовать и т.п. Современная теория нейронных сетей способна на некоторые из них давать строгие ответы, а на другие отвечать на основе богатого опыта. Строгие результаты основаны на (а) теореме Колмогорова об аппроксимации функций сигмоидами, (б) ряде теорем, показывающих, что многослойные персептроны являются универсальными аппроксиматорами (см. ниже) и (с) методах математической статистики. Очень хорошее описание статистического взгляда на нейронные сети можно найти в работах [16, 25, 81].

2.2 Нейросети в computer science

Это направление связано главным образом с логикой и вычислительными возможностями нейронных сетей. Отображение $X \rightarrow Y$, порождаемое нейросетью, можно рассматривать как способ организации вычислений. В этом плане основное отличие нейронных сетей от обычных компьютеров в том что они (1) могут работать с очень высокой степенью параллелизма, (2) вместо программирования, они обучаются на примерах и (3) они способны распознавать образы только по какой-либо их части.

В то же время между нейросетями и компьютерами нет жесткой границы. Начало истории искусственных нейронных сетей часто связывают со знаменитой работой Маккаллока и Питтса, опубликованной в 1943 году [63]. В ней показано, что простые сети пороговых элементов, иногда называемых нейронами Маккаллока-Питтса, могут работать как логические элементы, а значит из них можно составить большую сеть, которая будет работать как универсальный компьютер. С другой стороны, нейронные сети обычно реализуются как программы для классических компьютеров.

Другим интересным аспектом, связанным главным образом с исследованиями в области

искусственного интеллекта, является особый вид обучения. При обработке данных обычно известно значение выходного сигнала, которое должно соответствовать входным данным. Разницу между заданным и получающимся значением на выходе можно использовать для необходимой коррекции весов — это т.н. обучение с учителем. В задачах искусственного интеллекта, таких как управление перемещением робота, необходимо научиться правильным действиям в заданном "состоянии" (при данной комбинации показаний сенсоров) когда правильный ответ заранее неизвестен, известен только нужный окончательный результат и отклонение от него всей ситуации в целом, сложившейся в результате множества действий. Правильные действия в каждой конкретной ситуации можно найти только в процессе множества проб и ошибок. Такой путь обучения называется "обучение поощрением" (Reinforcement Learning) или "обучение с критиком". Оно основано на сигналах поощрения в случае успеха или наказания при его отсутствии. Первая попытка построения искусственной нейронной сети с поощрительным обучением была предпринята М. Минским еще в 1954 году, однако существенный прогресс в развитии методов обучения поощрением был достигнут лишь в 1980-1990 годах [83]. Важной чертой этого метода является интеграция нейросети в большую систему, при этом обучение протекает как взаимодействие обучающегося "агента" со своим окружением. Мы рассмотрим эти вопросы подробнее в разделах 5 и 6.

Хорошее изложение основ нейронных сетей с позиции computer science можно найти, например, в [7].

2.3 Искусственные нейронные сети и биология

Вопросы, связанные с вычислительными возможностями и историей нейронных сетей приводят к началам теории нейронных сетей, к биологическим аналогиям, вопросам теории восприятия (cognitive science) и теориям работы мозга (см., например, [78]). Другими словами, мы приходим к биологическому взгляду на искусственные нейронные сети. Прекрасное описание этой точки зрения можно найти, к примеру, в обзорах Фримана [34, 33].

Исследования свойств мозга в физиологии и психологии имеют значительно более давнюю историю, чем нейронные сети. Для биологии методы теории нейронных сетей — только один из многих возможных подходов. Его основная задача — моделирование и объяснение результатов физиологических экспериментов или углубление наших представлений о возможных принципах работы мозга. Подобное моделирование позволяет проверять биологические гипотезы, которые не поддаются проверке иными способами [34].

Вообще говоря, вся теория нейронных сетей обязана своим появлением открытиям в нейронауке. В 50-е годы возникли большие надежды на то, что вскорости удастся смоделировать основные функции мозга. Они возникли благодаря недооценке сложности мозга: первые оценки числа нейронов в мозге давали значения порядка $\sim 10^3$ [34]. Позднее было установлено, что мозг содержит около 10^9 нейронов, каждый из которых имеет большое число связей, до 10^5 . Задача моделирования функций всего мозга в этой связи отодвинулась на неопределенный срок.

Работы по моделированию биологических структур мозга начались с воспроизведения свойств единственного нейрона или его частей. Надо заметить, что биологический нейрон значительно сложнее формальных нейронов ИНС. Например, модель распространения электрического сигнала вдоль аксона (проводника выходного сигнала нейрона) включает

два уравнения в частных производных, а сравнительно простое описание динамики нейрона требует дифференциальных уравнений с запаздыванием (см., например, ссылки в [34]). Следующий уровень описания — модели групп нейронов или небольших структур мозга [34]. В этом случае основная задача состоит в построении сети, которая воспроизводила бы особенности, наблюдаемые в реальных физиологических экспериментах. Подобные модели могут быть абсолютно непохожи на те, что используются в аппроксимирующих нейросетях. Кроме того, в случае решения аналогичных задач, биологические сети оказываются гораздо сложнее.

Важным аспектом математической биологии нейронных сетей является классификация моделей, поиск наиболее общих их форм и принципов их построения. Существенный прогресс на этом пути был достигнут с использованием методов теории бифуркаций [53].

Многочисленные физиологические наблюдения демонстрируют сложность поведения мозга. Вполне вероятно, что оно является хаотическим [34, 8, 9, 38, 41]. Но почему? Какие преимущества в обработке информации может давать хаос? Необходима ли системам, обрабатывающим информацию, своя собственная внутренняя динамика и каковы в таком случае должны быть принципы их работы? Эти вопросы относятся уже к области нелинейной динамики.

2.4 Взгляд нелинейной динамики

С точки зрения нелинейной динамики, наиболее интересны общие принципы динамики исследуемой системы. Это может быть не слишком существенно для многих конкретных деталей, но зато дает основу для общего взгляда на различные типы нейронных сетей.

Во-первых, мы можем сразу разделить все типы нейронных сетей на две большие категории — "функции" и "динамические системы". Вторая категория для нас более интересна, хотя про первую мы скажем в дальнейшем несколько слов. Динамические системы могут быть *консервативными* или *диссипативными*. Все известные нам динамические сети являются диссипативными. Диссипация играет важную роль в распознавании образов. Она исключает шум или несущественные детали входных данных и превращает искаженный образ в "правильный". Процедура распознавания образов, таким образом, рассматривается как процесс стремления к аттрактору динамической системы. У консервативных систем аттракторов нет, а потому они не могут работать по этой схеме. Тем не менее, в разделе 7 мы рассмотрим пример консервативной гамильтоновой системы, реализующей алгоритм распознавания образов. Подобные системы могут представлять интерес с точки зрения квантовой обработки информации.

Аттракторы динамической системы представляют собой выход (Y) нейронной сети. Входной сигнал X можно подавать двумя основными способами: как начальные данные либо как параметры динамической системы. Комбинирование этих путей в принципе возможно, но ни одна подобная сеть нам неизвестна. Таким образом, нелинейная динамика позволяет предложить следующую классификацию нейронных сетей соответственно способу организации отображения $X \rightarrow Y$.

1. Сети-функции. Выходной сигнал получается в результате ряда явно заданных преобразований входного. Основные этапы преобразования представляют собой нейроны и веса. Подобные нейронные сети являются чаще всего обычными функциями нескольких переменных, зависящими от параметров: $Y = F(X, w)$, причем вид функции F

известен (явные сети). Но, вообще говоря, определение функции может включать рекурсивные "петли". Некоторые из выходных или промежуточных сигналов могут снова подаваться на вход, так что результатом может быть неявная функция, например, $Y = F(X, Y, w)$ (рекуррентные сети). Хорошо известными примерами сетей-функций могут служить (а) многослойные перцептроны и (б) сети с радиальными базисными функциями [16, 45].

2. Динамические сети первого типа: X = начальные данные, Y = конечное состояние (аттрактор). Динамика сети может описываться (i) обыкновенными дифференциальными уравнениями, (ii) отображениями либо (iii) клеточными автоматами. Заметим, что в области притяжения аттрактора конечное состояние системы Y в отображении $Y = F(X)$ не зависит от X — все притягивается к одному и тому же аттрактору. Таким образом, данный тип сетей представляет интерес только когда аттракторов несколько. Поэтому мы будем называть их *многоаттракторными сетями*. Для того, чтобы отображение F было задано правильно, то есть чтобы правильно обучить сеть, необходимо разместить аттракторы системы в нужных местах и обеспечить отсутствие "ложных" аттракторов. Наиболее известными сетями данного типа являются сети Хопфилда, BSB (brain state in a box), синергетический компьютер Хакена [49, 45, 27, 42].
3. Динамические сети второго типа: X = параметры динамической системы, Y = аттрактор. Как и в предыдущем случае динамическая система может иметь различный тип, но аттрактор для каждого набора входных данных аттрактор должен быть единственным. Наличие нескольких аттракторов может испортить работу системы. Роль входных данных заключается в манипулировании этим единственным аттрактором, с тем чтобы в результате получалось необходимое отображение. Поэтому мы будем использовать термин *сети с управляемым аттрактором*. Хорошо известными примерами таких сетей могут служить сети Гроссберга с адаптивным резонансом, сеть Хопфилда-Танка для задач комбинаторной оптимизации и модель Фримана обонятельной луковицы [18, 99, 51].
4. Гамильтоновы нейронные сети. Сети этого типа в контексте обработки информации практически не рассматривались в литературе.

С точки зрения нелинейной динамики, наиболее интересны такие свойства искусственных нейронных сетей как сложность динамики, устойчивость и предсказуемость. В частности, представляет интерес вопрос, способен ли динамический хаос улучшать работу нейронной сети и способен ли он играть какую-либо существенную роль в обработке информации? Основная цель данной статьи как раз и состоит в обсуждении этого вопроса. Но сначала мы рассмотрим более детально принципы работы некоторых важнейших типов нейронных сетей.

3 Основные типы искусственных нейронных сетей

В этом разделе мы кратко рассмотрим несколько основных типов нейронных сетей с позиций нелинейной динамики. Мы не ставили своей задачей сделать всеобъемлющий обзор,

подобный, например, [45]. Мы лишь хотим отметить некоторые черты каждого типа сетей, важные с динамической точки зрения.

3.1 Сети-функции

Рассмотрим простой пример аппроксимации функции $Y = f(X)$ сигмоидной функцией $\tanh(x)$ по значениям $Y_i = F(X_i)$ в нескольких точках $X_i \in [X_{01}, X_{02}]$. Поскольку область значений гиперболического тангенса — это отрезок $[-1, 1]$, прежде всего необходимо сделать так, чтобы значения $F(X_i)$ на $[X_{01}, X_{02}]$ ему принадлежали. Этого несложно добиться линейным преобразованием (предобработкой) Y_i , которую мы будем предполагать выполненной $Y_i \in [-1, 1]$. Затем необходимо выбрать вид аппроксимирующей функции $Y = F(X, \mathbf{w})$, где \mathbf{w} — набор параметров. Например, если зависимость близка к линейной, можно использовать функцию $Y = \tanh(w_1 X + w_2)$. Затем необходимо "обучить" эту аппроксимацию на примерах, т.е. найти наилучшие значения \mathbf{w} по известным парам X_i, Y_i . Подбор параметров можно осуществлять путем минимизации функционала ошибки $E(\mathbf{w}) = \sum_i (Y_i - F(X_i, \mathbf{w}))^2$.

Как всё это связано с динамикой? Расчет результата вообще не содержит никакой динамики. Процесс минимизации может принимать форму динамической системы, скажем, $\dot{\mathbf{w}} = -\nabla E(\mathbf{w})$, но любое усложнение этой динамики воспринимается как серьезный недостаток.

Тем не менее, динамика немедленно возникает как только мы попытаемся рассматривать *невявные* функции, например, $Y = \tanh(w_1 X + w_2 Y + w_3)$. Подобные невявные представления обычно более гибки, но они требуют специальной процедуры для поиска Y . Обычно это делается путем преобразования обычного уравнения в динамическую систему, например, $\dot{Y} = -Y + \tanh(w_1 X + w_2 Y + w_3)$. Разумеется, в данном простом уравнении никакой сложной динамики быть не может, однако в его многомерных аналогах она вполне возможна. Однако она по-прежнему рассматривается как серьёзный недостаток, поскольку основная задача — получить единственное нужное значение Y — остаётся невыполненной.

Многослойные перцептроны являются обобщением этого примера. Они включают несколько "сигмоидных" нейронов, состояния которых мы обозначим через x_i . Нейрон i получает сигналы от других нейронов через соединения с весами w_{ij} , а также внешний входной сигнал X_i . Сеть в целом выполняет преобразования вида

$$x_i = f \left(\sum_{j=1}^N w_{ij} x_j + \theta_i \right) + X_i. \quad (1)$$

f часто называют функцией активации, и обычно это сигмоидная функция.

Обобщением явных функций $Y = F(X, \mathbf{w})$ является класс явных сетей с прямым распространением сигнала (feed-forward networks), для которых $w_{ij} = 0$ при $j \geq i$, так что сигнал распространяется только от нейронов с меньшими номерами к нейронам с большими. Первый нейрон получает только сигнал извне, второй — извне и, возможно, от первого и т.д. Для обучения таких сетей применяется метод, называемый обратным распространением ошибки. Чтобы корректировать веса при обучении, необходимо сначала найти вспомогательную "погрешность" e_i для каждого нейрона. Можно показать, что e_i удовлетворяют системе линейных уравнений с верхней треугольной матрицей (в то время

как w — нижняя треугольная). Поэтому x_i рассчитывают поочередно с 1 до N , а затем e_i от N до 1, после чего рассчитывают поправки весов Δw_{ij} .

Обобщением неявных функций $Y = F(X, Y, \mathbf{w})$ являются рекуррентные нейронные сети, со связями в обоих направлениях. Для них ограничений на структуру матрицы w_{ij} нет. При решении системы (1) приходится использовать какие-либо итерационные схемы, то есть превращать систему нелинейных уравнений в динамическую систему. Например, вместо $x = f(wx) + X$ рассматривают систему $\dot{x} = -x + f(wx) + X$, неподвижные точки которой дают решения исходных уравнений. Метод обратного распространения ошибки, связанный с вычислением вспомогательных погрешностей e_i , был обобщен и на рекуррентные сети [74, 45], однако уравнения для e_i более не имеют столь простого вида.

Использование рекуррентных персептронов может сталкиваться с трудностями, поскольку (а) решение может не существовать, (б) может быть неединственным, либо (с) итерационная схема может приводить к нетривиальному временному поведению вместо сходимости к неподвижной точке.

Другим широко известным типом сетей-функций являются сети с **радиальными базисными функциями**. Для них выбирается скалярная *базовая функция* $\varphi(r)$ одной переменной, а также набор узлов X_{0k} , $k = 1, \dots, M$. Аппроксимация ищется в виде $F(X) = \sum_{k=1}^M A_k \varphi(\|X - X_{0k}\|)$. Значения A_k можно найти минимизируя функционал ошибки $E(A) = \sum (Y_i - F(X_i))^2$. Решение существует в широком классе базисных функций. Динамика, как легко заметить, отсутствует.

Тем не менее описанные типы нейронных сетей тесно связаны с нелинейной динамикой. Они используются для аппроксимации неизвестных уравнений движения динамических систем по временным рядам. Временной ряд поставляет пары X_k (текущее состояние динамической системы) и Y_k (следующее состояние), а задача заключается в аппроксимации зависимости $Y = F(X)$. Затем построенная аппроксимация может быть использована для предсказания временного ряда [96] (существует также особый тип персептронов, ориентированный на обработку временных рядов и версия методики обучения, называемая “backpropagation through time”) [45].

3.2 Многоаттракторные сети

3.2.1 Простейший пример

Рассмотрим простейший пример градиентной системы, потенциал которой имеет две ямы,

$$\dot{x} = x - x^3, \quad x(0) = X.$$

Если в качестве выходного значения взять $Y = x(\infty)$, то при $X < 0$ мы получим $Y = -1$, а при $X > 0$ — $Y = 1$, то есть $Y = \text{sgn}(X)$. Эта динамическая система распознаёт знак входных данных и разбивает их на два класса. Или же мы можем сказать, что в систему заложены два “образа” (+1 и -1, один бит), а начальные данные представляют собой “зашумлённый” или “искаженный” образ. Система же распознаёт входные данные как один из двух запомненных образов.

Все прочие многоаттракторные сети функционируют аналогично: (1) запомненные образы соответствуют аттракторам динамической системы; (2) классификация или распознавание выполняется в соответствии с тем, в области притяжения какого аттрактора

окажутся начальные данные; и (3) процесс сходимости к аттрактору рассматривается как распознавание образа. Эта интерпретация подразумевает, что аттракторов должно быть больше одного.

3.2.2 Сети Хопфилда

Этот тип многоаттракторных сетей известен наиболее широко. В них каждый нейрон связан с каждым, а уравнения движения имеют вид

$$\hat{x}_i = f\left(\sum_j w_{ij}x_j + \theta_i\right), \quad (2)$$

где \hat{x}_i — значение переменной x_i на следующем временном шаге. Важной особенностью сетей Хопфилда [49, 50] (в отличие от некоторых очень близких аналогов) является асинхронность эволюции: на каждом шаге по времени изменяется состояние только одного нейрона. Заметим, что иногда термин "сеть Хопфилда" используется по отношению и к более широкому классу моделей с синхронной эволюцией или когда часть связей отсутствует.

В простейшем варианте сети нейроны являются "двоичными", т.е. x_i принимает только два значения, ± 1 , а f — это "ступенька" $f(u) = \text{sign}(u)$, $\theta_i = 0$.

Обучение сети осуществляется путём запоминания двоичных образов $\xi^{(k)}$, $\xi_i^{(k)} = \pm 1$, в соответствии с "правилом Хебба",

$$w_{ij} = \frac{1}{N} \sum_{k=1}^M \xi_i^{(k)} \xi_j^{(k)}, \quad i \neq j, \quad w_{ii} = 0, \quad (3)$$

$i, j = 1, \dots, N$. Это означает, что w_{ij} увеличиваются если $\xi_i^{(k)} = \xi_j^{(k)}$ и уменьшаются в противоположном случае.

Интересной особенностью сетей этого типа является существование функционала энергии или функции Ляпунова

$$E = - \sum_{i,j=1}^N w_{ij}x_i x_j.$$

Нетрудно показать, что на каждом шаге по времени E не возрастает: если не меняется x_i , остаётся неизменным и E . Если x_i меняется ($\hat{x}_i = -x_i$), то $\Delta E < 0$. Поскольку E ограничено снизу, эволюция должна закончиться в точке равновесия, где энергия достигает минимума.

Основным недостатком сетей этого типа является небольшой объём памяти. Если образы не выбраны специальным образом, так чтобы они были практически ортогональны, как правило, сеть способна правильно распознавать около $0.14N$ образов. Если попытаться запомнить большее количество, возникающие аттракторы перестают соответствовать запоминаемым образам. Было предложено несколько способов борьбы с ложными образами, см. например, [45, 37, 72, 66] (немонотонность f , введение в правило обучения обратной корреляционной матрицы образов, специальная десимметризация матрицы весов; лучший достигнутый результат — методика Э. Гарднер [37], дающая запоминание до $\sim 2N$ образов, правда, не вполне произвольных). Мы ещё вернёмся к этой проблеме, когда будем обсуждать возможные роли хаоса в нейронных сетях.

Заметим, что в модели Хопфилда используется симметричная матрица весов. В случае несимметричной матрицы в синхронной модели возможно колебательное поведение, но в настоящее время оно практически не использовалось в целях распознавания образов.

3.2.3 BSB ("Brain state in a box") — "Мозг в ящике"

Несмотря на столь претенциозное название, эти сети не сложнее предыдущей. Для них (см. например [43, 45]) уравнения движения имеют вид

$$\dot{x}_i = f(x_i + \beta \sum_{j=1}^N W_{ij} x_j),$$

а функция $f(x) = x$ при $x \in [-1, 1]$ и $f(x) = \text{sgn}(x) = \pm 1$ если $x \notin [-1, 1]$, матрица W должна быть положительно определённой. Как следует из уравнений, состояние системы всегда заключено в N -мерный куб $[-1, 1]^N$. Этот факт весьма существен для работы модели. Несложно показать, что вершины куба $|x_i| = 1$ будут устойчивыми притягивающими состояниями при условии $W_{ii} \geq \sum_{j \neq i} |W_{ij}|$ [45]. Это условие гарантирует устойчивость вершин и заодно является достаточным для положительной определённости W . Однако для матриц, возникающих на практике, лишь некоторые из вершин оказываются устойчивы. По этой причине модель BSB успешно использовалась в задачах классификации, когда каждому кластеру соответствовала одна из устойчивых вершин куба.

Матрицу W можно построить адаптивным образом по некоторой учебной выборке $x^{(k)}$ при помощи итерационного алгоритма

$$W_{k+1} = W_k + \epsilon(x^{(k)} - W_k x^{(k)}) x^{(k)T} = W_k(1 - \epsilon P_{x_k}) + \epsilon P_{x_k},$$

где $P_x = x x^T$ — проектор на направление вектора x . Если эту процедуру повторять бесконечно долго для одного и того же вектора x , она сойдётся к матрице W_∞ , для которой x будет собственным вектором с собственным значением $\lambda = 1$, $W_\infty x = x$. Если для векторов обучающей выборки можно выделить несколько наиболее важных направлений, эта процедура способна их найти, подобно анализу главных компонент. После получения матрицы W модель BSB делает результаты классификации видимой: устойчивые вершины отвечают кластерам для обучающей выборки и позволяют проводить классификацию последующих данных.

3.2.4 Самоорганизующиеся сети Кохонена (self-organizing maps, SOM)

Сети этого типа могут быть построены несколькими способами: как многоаттракторные сети, как сети с управляемым аттрактором, а если "разрешить" использование несколько необычной функции $\text{arg max} \{x_1, x_2, \dots, x_N\}$, (возвращающей не сам наибольший элемент x_k , а только его номер k), то и как сети-функции вообще без динамики. Из динамических реализаций проще оказывается многоаттракторный вариант, который мы и рассмотрим. Основное отличие этой сети от прочих динамических сетей состоит в том, что её обучение никак не влияет на её динамику. Обучение связано лишь с предобработкой входных данных.

Сеть состоит из N нейронов с конкурентной динамикой: каждый активный нейрон пытается подавлять активность прочих нейронов, так что в конце концов только один

нейрон, тот, у которого значение $x(0)$ было наибольшим, остаётся активным, $x_k(\infty) = 1$, а все прочие $x_i(\infty) = 0$. Аттрактор состоит из вектора, включающего $N - 1$ ноль и единственную единицу. Номер k победившего нейрона служит индикатором распознанного образа или кластера входных векторов. Конкретная форма уравнений движения не очень важна. Например, уравнения могут иметь вид $\dot{x}_i = ax_i(1 - \sum_{j=1}^N x_j^2)$. Динамика работает только как "проявитель" того, у которого нейрона входной сигнал больше. В прикладных задачах динамику часто вообще опускают и заменяют выражением $k = \arg \max_j x_j(0)$. Пример описанной выше сети с подробным анализом динамики можно найти, к примеру, в работе [10].

Все свойства к распознаванию и классификации у сетей данного типа коренятся в способе получения $x_i(0)$ из входного сигнала X . Связи между нейронами не зависят от набора образов, которые должны распознаваться. Но кроме внутренних связей каждый нейрон x_i обладает n входными связями w_{ji} , $j = 1, \dots, n$, $n = \dim X$, которые и используются при получении $x_i(0)$. Например, так, $x_i(0) = R - \sum_j (X_j - w_{ji})^2$. Здесь удобно обозначить i -й столбец матрицы w_{ji} как вектор \mathbf{w}_i , тогда $x_i(0) = R - \|X - \mathbf{w}_i\|^2$. Очевидно, что в таком случае победителем окажется нейрон, у которого \mathbf{w}_i ближе всего к X . Если все векторы \mathbf{w}_i нормированы, $\|\mathbf{w}_i\| = 1$, то $x_i(0) = R - X^2 - \|\mathbf{w}_i\|^2 + 2(X \cdot \mathbf{w}_i)$. Поскольку лишь последний член зависит от i , то можно просто положить $x_i(0) = (X \cdot \mathbf{w}_i)$ с тем же самым результатом распознавания. Последняя форма записи более "нейронна": нейрон получает сигналы X_j через связи с весом w_{ij} .

Если сеть должна распознавать до N известных образов $X^{(i)}$, можно просто положить $\mathbf{w}_i = X^{(i)} / \|X^{(i)}\|$. Сеть будет проверять, который из известных образов ближе всего к входному сигналу и "включать" соответствующий нейрон, устанавливая соответствующий x в 1. Если требуется не просто индикатор структуры, а ещё и сама структура, можно сгенерировать и n -мерный выходной сигнал. Для этого необходимо каждый из нейронов снабдить выходными связями $\mathbf{w}_i^{\text{out}}$ и генерировать выходной сигнал сети $Y = \sum_{i=1}^N x_i \mathbf{w}_i^{\text{out}}$. Так как только один из x_i отличен от нуля, скажем, $x_k = 1$, то выход всей сети окажется равным $\mathbf{w}_k^{\text{out}}$. Если требуется воспроизвести запомненный образ, как это было в сетях Хопфилда, следует просто положить $\mathbf{w}_i^{\text{out}} = \mathbf{w}_i = X^{(i)}$. Однако в данном случае свободы больше и можно положить выход равным произвольному вектору $\mathbf{w}_i^{\text{out}} = Y^{(i)}$. Такая сеть будет осуществлять отображение $X \rightarrow Y^{(k)}$ если $k = \arg \min_i \|X - X^{(i)}\|$. Области притяжения для переменных x_i отвечают областям в пространстве входных сигналов X . Эти области иногда называют ячейками Вороного или Дирихле: если X принадлежит k -й ячейке, то ближайшим из $X^{(i)}$ к нему будет $X^{(k)}$ [59].

Самоорганизующиеся сети — это фактически способ обучения подобной архитектуры разбивать входные данные на "самоорганизующиеся кластеры", то есть способ создания ячеек Дирихле с учетом структуры входных данных. Допустим, что у нас есть M образов $X^{(i)}$, которые необходимо разбить не более чем на N кластеров. Очевидно, что входные веса \mathbf{w}_i необходимо выбирать в соответствии с данными. Делается это при помощи следующей итерационной процедуры. Векторы $X^{(i)}$ поочередно предъявляются сети. После определения нейрона-победителя x_k его входные связи \mathbf{w}_k корректируются: $\hat{\mathbf{w}}_k = \mathbf{w}_k + \epsilon (X^{(i)} - \mathbf{w}_k)$. То есть \mathbf{w}_k чуть-чуть сдвигается к $X^{(i)}$. После того, как данная процедура повторяется порядка $\sim 10^4$ раз, веса \mathbf{w}_k обычно дают хороший набор кластеров для классификации входных образов [59, 45].

В такой простой версии алгоритма, соседние нейроны не обязательно соответствуют близким кластерам. Поэтому Кохонен предложил несложную процедуру для большего соответствия положений кластеров положениям нейронов. Все нейроны располагаются в узлах одно- или двумерной решетки, так что у каждого нейрона появляется естественная окрестность. Далее при обучении веса корректируются не только у победившего нейрона, но и у его нескольких ближайших соседей, так что у соседних нейронов веса оказываются более близкими друг к другу. Далее с течением времени размер окрестности (то есть число соседних нейронов, у которых изменяются веса) сокращается, так что в конце обучения изменения затрагивают только один нейрон-победитель. В результате близким нейронам обычно соответствуют близкие кластеры. Зачастую, когда входные векторы X двумерны, результаты обучения изображают как распределение векторов \mathbf{w}_i на плоскости со связями, отражающими структуру нейронов. Интересно, что если нейроны располагаются в виде одномерной решётки, а распределение кластеров входных данных имеет большую размерность, то одномерная структура векторов \mathbf{w} пытается заполнить область более высокой размерности подобно первым шагам построения фрактальной кривой Пеано. Благодаря этому свойству сети Кохонена часто называют "топологическими сетями".

Технику организации кластеров в пространстве векторов иногда называют "векторным квантованием" (vector quantizing). Её приложения весьма многочисленны [59]. Наиболее впечатляющим среди них, видимо, является созданная Т. Кохоненом в середине 80-х годов печатающая машинка с автоматическим распознаванием устной речи: сеть была обучена соответствию между звуками речи и буквами, которые следует печатать в ответ на эти звуки [60, 59].

Идея сетей Кохонена, дающих самоорганизующееся отображение вида вектор \rightarrow номер кластера, была обобщена Р. Хехт-Нильсеном на самоорганизующиеся отображения вида вектор \rightarrow вектор [47]. Сеть получила название "counterpropagation network". В ней, фактически, проводится кластеризация векторов, состоящих одновременно из образцов как входных, так и выходных сигналов, т.е. известных пар $X^{(i)}, Y^{(i)}$. Получающееся отображение $X \rightarrow Y$ кусочно постоянно внутри каждой из ячеек Вороного-Дирихле, и для всех X внутри данной ячейки Y равно значению в центре соответствующей ячейки в Y -пространстве.

3.2.5 Синергетический компьютер и динамические системы с многими потенциальными ямами

Синергетический компьютер был предложен Г. Хакеном [42]. Он напоминает непрерывную версию сетей Хопфилда [50], однако с некоторыми изменениями. Для запоминания образов также используется матрица, сформированная по правилу Хебба. Однако вместо сигмоидной функции, обеспечивающей ограниченность правых частей уравнений движения, для стабилизации образов используются полиномиальные члены высших порядков.

Другим способом построения динамической системы с неподвижными точками в нужных местах послужил специальный выбор потенциала градиентной системы вида $U(\mathbf{x}) = \sum A_i V(\mathbf{x} - \mathbf{x}_i)$. Подбирая параметры, можно решить проблему взаимной интерференции аттракторов. Этот способ был предложен в начале 90-х, однако распространения не получил. Видимо, с одной стороны в нем совсем уже нет никаких нейронных аналогий, а кроме того он лишен мистического очарования сетей Хопфилда, где формирование аттракторов

— процесс далеко не очевидный.

Мы описали лишь некоторые, как нам казалось, наиболее важные типы многоаттракторных сетей. Хотя для их "классических" версий сложная динамика совершенно несущественна, ниже мы рассмотрим несколько попыток преодолеть с её помощью некоторые недостатки сетей, например, ложную память в сетях Хопфилда.

3.3 Сети с управляемым аттрактором

3.3.1 Простой пример

Рассмотрим уравнение

$$\dot{x} = -x + X.$$

Это градиентная система с потенциалом $U(x) = x^2 - Xx$. Входные данные этот потенциал изменяют. Для любых начальных данных $x(0)$ конечное состояние $x(\infty) = X$. Следовательно, система осуществляет тождественное преобразование $Y = X$.

Разумеется, тем же путём можно реализовать и другие функции. Скажем, система $\dot{x} = -x + \tanh(X)$ порождает сигмоидный выходной сигнал. Другой (и возможно более интересный) результат может быть достигнут при помощи системы

$$\dot{x} = -\frac{ax}{1-x^2} + X.$$

Она порождает отображение

$$Y = \frac{2X}{a + \sqrt{a^2 + 4X^2}},$$

то есть тоже почти сигмоидную функцию. Заметим, что при малых a эта функция близка к ступеньке $Y = \text{sign}(X)$, и может быть использована для целей классификации или распознавания, подобно случаю многоаттракторных сетей.

Следует, однако, заметить, что интерпретация $x(\infty)$ как значения функции $F(X)$ возможна только при условии, что при каждом X динамическая система имеет единственный аттрактор, который должен быть неподвижной точкой. В противном случае у F может оказаться несколько значений при одном и том же X или же предельное значение $x(\infty)$ может не существовать. Обычно разработчики нейронных сетей стараются избегать подобных ситуаций и используют теоремы, гарантирующие существование устойчивой неподвижной точки, например, известную теорему Коэна-Гроссберга [39]. Как правило, существование функции Ляпунова для уравнений динамики сети и отсутствие сложной временной динамики считается большим достоинством.

Некоторые сети можно построить обоими путями, и как многоаттракторную сеть, и как сеть с изменяемым аттрактором, например, сети Кохонена, рассмотренные в предыдущем разделе. В последнем случае входной сигнал X должен постоянно подаваться на вход сети и служить в качестве параметров в уравнениях движения. Другими известными примерами сетей с изменяемым аттрактором являются сети Гроссберга с адаптивным резонансом и модель обонятельной луковицы, предложенная В. Фриманом. Последняя демонстрирует сложную временную динамику и мы обсудим её в разделе 4.

3.3.2 Сети с адаптивным резонансом (ART — adaptive resonance theory)

Подобно сетям Кохонена, базовая ART сеть получает на входе X и выдаёт на выходе индикатор соответствующего кластера. Некоторые обобщения приведены в работе [18]. Согласно этой работе, правильное функционирование сети ART требует довольно тщательной настройки. Поэтому приводимое обсуждение — это только краткий обзор некоторых идей, лежащих в основе ART, а не рекомендация по построению таких сетей.

Нейроны в сети располагаются тремя слоями: входной слой, на котором сигнал X фиксирован, преобразованный входной слой F_1 (состояния его нейронов будем обозначать x_i) и выходной слой F_2 (с элементами y_j). Поскольку как и в случае SOM, выход сети обозначает номер кластера, только одно из y_j в конце концов должно остаться ненулевым. Обработка информации в сети протекает по следующей схеме: $X \rightarrow F_1 \leftrightarrow F_2 \rightarrow \text{выход}$.

В наиболее компактном и удобном виде уравнения движения приведены в [80]. Мы рассмотрим только вариант ART1, разработанный для классификации двоичных входных данных, когда компоненты X могут принимать только значения 0 или 1. Динамика на слое F_1 описывается уравнениями

$$\frac{dx_i}{dt} = -x_i + (1 - A_{11}x_i) \left(X_i + \sum_j T_{ij}f(y_j) \right) - (A_{12} + A_{13}x_i) \sum_j f(y_j).$$

Первый член в правой части гарантирует, что в отсутствие входного сигнала $x \rightarrow 0$. Второй член обеспечивает ограниченность x при наличии входного сигнала. Сумма $X_i + \sum_j T_{ij}f(y_j)$ приводит к тому, что x может существенно вырасти только если входной сигнал X соответствует образу, запомненному в матрице обратных ($F_2 \rightarrow F_1$) связей T — т.н. "ожидаемый вход", отвечающий текущему состоянию слоя F_2 . Третий член позволяет подавить существующее состояние x когда отсутствует соответствие между входом и ожиданием. Функция f — это сигмоида: $f(x) = 0$ при $x \leq 0$, а при $x > 0$ она растёт и стремится к 1.

Уравнения для слоя F_2 выглядят аналогично,

$$\frac{dy_j}{dt} = -y_j + (1 - A_{21}y_j) \left(f(y_j) + \sum_i B_{ij}f(x_i) \right) - (A_{22} + A_{23}y_j) \left(\sum_{k \neq j} f(y_k) + r_j \right).$$

Второй член в правой части, как и в сети Кохонена, возбуждает нейроны верхнего слоя в соответствии в матрицей прямых связей B_{ij} плюс некоторое самовозбуждение, а третий член ограничивает активность нейронов и включает особый член сброса, о котором мы скажем ниже.

Динамика обучения для матриц B и T довольно проста и напоминает случай сетей Кохонена:

$$\tau \frac{dT_{ij}}{dt} = f(y_j) (f(x_i) - T_{ij}), \quad \tau \frac{dB_{ij}}{dt} = f(y_j) \left(\frac{f(x_i)}{1 + |F_1|} - B_{ij} \right),$$

где $|F_1|$ — активность слоя F_1 (количество активных нейронов в слое). Согласно этим уравнениям, обучение происходит только для тех связей, которые идут к активному нейрону y_j , а запоминается образ, возникающий в слое F_1 . Уравнения выражают хорошо известное правило Хебба.

Рассмотрим теперь как работает сеть. Входной образ X активирует слой F_1 , а тот в свою очередь через матрицу связей B порождает некоторую структуру в слое F_2 , где один из y_j оказывается активирован. Образ, запомненный в матрице T , отвечающий $y_j = 1$ порождает сигнал обратной связи для слоя F_1 . Если эти два сигнала соответствуют друг другу, сеть остаётся в этом состоянии и подстраивает к нему матрицы B и T . Если же входной и обратный (ожидаемый) образы недостаточно близки, запускается механизм поиска. Прежде всего, активируются переменные сброса r ,

$$\tau_r \frac{dr_j}{dt} = f(y_j) (Df(v|X| - |F_1|) - r_j).$$

Здесь v — так называемый ‘параметр бдительности’, который характеризует близость двух образов, а $|X|$ и $|F_1|$ обозначают количество активных нейронов в слоях X и F_1 . После активации r_j подавляет нейрон y_j в слое F_2 (для лучшей работы сети он должен оставаться в этом состоянии до конца поиска). Тогда влияние обратной связи временно исчезает, структура слоя F_1 восстанавливается и сеть начинает активировать другой нейрон в F_2 . Если его ожидаемый образ, записанный в T также не соответствует, он также подавляется и поиск продолжается до тех пор пока подходящий нейрон не будет найден. Это может оказаться и новый нейрон, для которого ожидаемый образ ещё не сформировался, тогда начнётся формирование нового кластера. Или же все нейроны в слое F_2 могут быть исчерпаны, тогда сеть неспособна классифицировать входной сигнал.

Соответствие образов в слоях F_1 и F_2 С. Гроссберг назвал адаптивным резонансом. Мы описали его механизм лишь в самых общих чертах. Существует несколько вариантов таких сетей: ART1 для двоичных образов, ART2 для ‘аналоговых’ сигналов, ART3, где слои F_1 и F_2 симметричны. Кроме того каждый вариант включает несколько версий. Заметим, что правильное функционирование сети требует весьма тонкой настройки всех параметров, детальное описание можно найти в работах [19, 20, 21, 39, 18, 80].

Интересной особенностью сетей ART1 и ART2 является то, что их можно активировать с обеих сторон, то есть слой F_2 в принципе можно сделать входным. Тогда два модуля ART2 можно объединить, так что первый получает входной сигнал и классифицирует его, а второй получает номер класса на свой слой F_2 и генерирует аналоговый выходной сигнал на своём F_1 . Такая комбинированная сеть получила название ARTMAP [22]. Подобно counterpropagation сети Хехт-Нильсена, такая конструкция даёт кусочно-постоянную аппроксимацию неизвестной векторной функции многих переменных.

3.3.3 Оптимизирующая сеть Хопфилда-Танка

Эта сеть была предложена в работе [51]. Уравнения движения сети совпадают с уравнениями непрерывной версии сети Хопфилда [50],

$$\frac{dx_i}{dt} = -\frac{1}{\tau} x_i + \sum_{j=1}^N w_{ij} f(x_j) + I_i, \quad (4)$$

где f — сигмоидная функция, w — матрица связей, а I_i — порог срабатывания нейрона i . Уравнения (4) описывают динамику скорейшего спуска $\dot{x} = -\nabla E$ для некоторого функционала энергии $E(f, w, I)$, а потому конечное состояние должно быть неподвижной точкой, отвечающей минимуму E . Обучение сети заключается в построении матрицы w и

сдвигов I специальным образом, чтобы поместить точку минимума в нужное место фазового пространства. Слабость метода состоит в том, что трудно гарантировать отсутствие ложных аттракторов. Ниже в разделе 4 мы увидим, как хаос позволяет избегать ложных минимумов.

Для задачи коммивояжера Хопфилд и Танк предложили специальный вид функционала энергии, зависящего от расстояний между городами d_{XY} , которые следует посетить. Следовательно, мы приходим к общему виду задачи об управляемом аттракторе с параметрами d_{XY} . Из-за ложных аттракторов решения, найденные сетью, могут не быть оптимальными, но, согласно [51], достаточно близки к ним.

3.3.4 Прочие сети

Существуют и другие сети с управляемым аттрактором. Например, в эту категорию попадает хорошо известная машина Больцмана, поскольку во время обучения часть её нейронов фиксируется и играет роль параметров, а не динамических переменных. Сюда же относится и модель обонятельной луковицы Фримана [35]. Некоторые сети этого типа мы рассмотрим в следующем разделе.

4 Сложная динамика как средство улучшения нейронных сетей

Искусственные нейронные сети зарекомендовали себя как универсальный инструмент в задачах аппроксимации, распознавания образов и т.п. Тем не менее, зачастую их применение связано с рядом трудностей. Скажем, для сетей-функций проблемы возникают при попытке аппроксимации слишком сложных функциональных зависимостей: слишком простая сеть неспособна дать правильной аппроксимации, а слишком сложная начинает усиливать шум (с этой проблемой в её простейшем виде хорошо знаком каждый, кто сталкивался с задачей выбора степени полинома при полиномиальной аппроксимации одномерной зашумлённой функции). Этот эффект был назван “overfitting” (“переаппроксимация”) и решение обычно ищется при помощи введения иерархических структур сети, получивших название модульных сетей и ансамблей сетей [82].

Для обычных динамических сетей, рассмотренных выше, типичными проблемами являются 1) ложная память, 2) слишком медленная сходимость к аттрактору и 3) неспособность воспроизводить реальную активность мозга.

Динамика в самом деле имеет определённый потенциал в области обработки информации. Например, очень интересная идея была предложена А.С. Дмитриевым: изображение запоминается в виде длинного цикла динамической системы, а каждый шаг цикла отвечает одному пикселу изображения.

Другой тип сетей со сложной динамикой возник из аналогии между нейроном или небольшой группой нейронов и осциллятором. Такая сеть представляет собой решётку связанных осцилляторов. Запись и обработка информации может осуществляться с использованием фазы осцилляторов. Такие сети осцилляторов могут использовать принципы работы сетей Хопфилда [52] или иных типов сетей, например, [69, 48].

Идея того, что нейронная сеть может работать в хаотическом режиме также высказывалась в ряде работ. Хороший обзор основных направлений в этой области на начало 90-х годов приведён в [34, 93], и мы не будем повторять то, что уже было сказано. Мы кратко скажем о некоторых моделях, отражающих основные направления в данной области. Потенциальная польза от хаоса вытекает из следующих его свойств [28, 40]:

1. *Локальная неустойчивость.* Она может быть полезна для устранения ложной памяти и избегания траекторий, ведущих в нежелательные области фазового пространства. Однако эти свойства хаоса могут быть полезны лишь в течение переходного периода, пока траектория не сойдётся к "истинному образу". После этого динамика снова должна стать устойчивой.

2. *Порождение информации.* Это свойство хаоса выглядит наиболее привлекательно. Тот факт, что хаотическая система ведёт себя неожиданным образом, может быть использован для поиска новых способов решения проблем. Иными словами, хаос может помочь в *освоении новых возможностей.* Однако современные нейронные сети неспособны использовать это свойство, поскольку их желаемое поведение должно быть полностью предсказуемым: они всегда должны давать один и тот же отклик на данный входной сигнал. Были попытки использовать хаос в алгоритмах случайного поиска минимума при обучении многослойных персептронов, однако детерминированные алгоритмы оказались лучше. Нейронные сети, способные осваивать возможности при обучении ещё только предстоит создать.

3. *Блуждание по аттрактору.* Это свойство тесно связано с неустойчивостью. Делались попытки использовать его как средство поиска нужного образа на хаотическом аттракторе. Траектория блуждает между различными образами, и идея состояла в том, чтобы использовать это свойство для сравнения входных данных с запомненными образами. Были получены кое-какие предварительные результаты, однако соответствующий алгоритм распознавания образов так и не был создан.

4. *Сходство со сложной динамикой мозга.* Ряд попыток был предпринят главным образом Фриманом и его сотрудниками [34, 31, 32, 99] для создания нейронной сети со структурой, напоминающей строение части мозга — обонятельной луковицы. Активность мозга сильно изменяется со временем, то же бывает справедливо и для хаотических систем. Оказалось, что поведение полученных моделей напоминает экспериментальные сигналы, снятые с реального мозга.

5. Наконец, хаос может не играть никакой особой роли: можно создать многоаттракторную систему с периодическим или хаотическим аттрактором вместо неподвижной точки [10]. Тип или номер аттрактора может рассматриваться как результат распознавания, хотя при этом может потребоваться специальная декодирующая система, преобразующая хаотический сигнал в информативный отклик сети.

Ниже мы рассмотрим несколько примеров хаотических нейросетей.

4.1 Переходный хаос как средство от ложной памяти

4.1.1 Хаотизация сети Хопфилда-Танка

Одним из недостатков модели Хопфилда-Танка было наличие множества локальных минимумов энергии. Для преодоления этой трудности ряд исследователей [87, 62] использовали

перехоный хаос и шум. Недавно Kwok и Smith [62] развили обобщённый подход для сетей такого типа. Мы, однако, ограничимся только одним примером для пояснения общей идеи.

Недавно Chen и Aihara [24] применили переходный хаос в сети, описываемой уравнениями

$$\begin{aligned} x_i(t+1) &= kx_i(t) + \alpha \left(\sum_{j=1, j \neq i}^N w_{ij} f(x_j(t)) + I_i \right) - z(t)(f(x_i(t)) - I_0) \\ z(t+1) &= (1 - \beta)z(t) \end{aligned} \quad (5)$$

где $z(t) \geq 0$ — член самовозбуждения, а $0 \leq \beta \leq 1$ — коэффициент затухания. Эти уравнения можно получить из уравнений аналогичных (4) введением дискретизации по времени. Благодаря последнему члену, содержащему $z(t)$, система может демонстрировать хаотическое поведение. Эволюция начинается с больших значений $z(t)$ чтобы обеспечить существование хаоса, а затем $z(t)$ уменьшается в соответствии с (5) и система получает возможность сойтись к нужному аттрактору. Таким образом, хаос помогает бороться с ложной памятью.

Другой пример сети с переходным хаосом для задач оптимизации предлагался в [23]. Сеть снова является отображением, но на этот раз с запаздыванием

$$x_i(t+1) = \sum_{j=1}^N w_{ij} f(x_j(t)) + I_i + g(x_i(t) - x_i(t-1)), \quad (6)$$

где f , как обычно, сигмоидная функция, а $g(x) = axe^{-b|x|}$. В стационарном состоянии последний член пропадает, поэтому неподвижные точки (6) те же, что и в модели Хопфилда-Танка (4).

Член g усложняет уравнения и приводит к хаотическому блужданию. Благодаря ему, ложные устойчивые состояния модели Хопфилда-Танка при численном моделировании становятся неустойчивыми, хотя траектории задерживаются на некоторое время вблизи них. Для того чтобы обеспечить сходимость к глобальному минимуму, авторы [23] предложили специальную схему изменения параметров a и b . Как показали эксперименты, сеть успешно справилась с задачей коммивояжера.

Модификация модели Хопфилда-Танка с аддитивным хаосом или шумом рассматривалась также в [44].

4.1.2 Сети хопфилдовского типа с переходным хаосом

Эта сеть была предложена в работе [55]. Идея заключалась в том, чтобы заменить простой пороговый хопфилдовский нейрон одномерной динамической системой $x_{t+1} = f(x_t, \mu)$ — нейроном со своей собственной внутренней динамикой.

Получающаяся сеть отображений управляется энергией системы E через параметр μ . Если энергия большая, динамика системы хаотическая, а когда энергия понижается, траектория отображения стремится к одной из неподвижных точек близких к ± 1 . Следовательно, во время переходного периода динамика системы хаотическая, а вблизи энергетического минимума она становится регулярной. Уравнения движения каждого "нейрона" имеют вид

$$x_i(t+1) = f(x_i(t), E_i), \quad (7)$$

где

$$F(x, E) = \{K(E)(x + |E|)\} \bmod 2 - 1, \quad K(E) = 2(1 + |E|)^{-1}. \quad (8)$$

(хаотическое поведение при $|E| < 1$ и регулярное в противном случае). Параметр E можно назвать локальной энергией системы

$$E_i(t) = \lambda \sum_{j=1}^N w_{ij} x_j(t).$$

Матрица w , как и в исходной модели Хопфилда, строится по правилу Хебба (3). Определение локальной энергии E_i включает параметр λ , который описывает взаимодействие нейронов: при $\lambda = 0$ сеть распадается на N независимых отображений, а при $\lambda = \infty$ получается обычная модель Хопфилда.

Численные эксперименты показывают, что при больших значениях λ система действительно ведёт себя как модель Хопфилда, но при меньших значениях (в примере, рассмотренном в [55], $\lambda \leq 5$) возникает существенная разница: почти все ложные образы становятся неустойчивы, и система либо остаётся в хаотическом состоянии в течение длительного времени, либо сходится к одному из запомненных образов. Таким образом, в данном случае хаос позволяет избавиться от ложных образов.

4.1.3 Решётка связанных отображений с нестационарными синхронными кластерами

Сети данного типа описаны в [54]. Они основаны на решётке связанных отображений с уравнениями движения

$$x_i(t+1) = (1 - \epsilon_i) f_i(x_i(t)) + \frac{\epsilon_i}{N} \sum_{j=1}^N f_j(x_j(t)), \quad (9)$$

$$f_i(x) = \alpha_i x^3 - \alpha_i x + x, \quad 2 \leq \alpha_i \leq 4. \quad (10)$$

Существует две версии системы: α -версия, когда $\epsilon_i = \epsilon$, α_i различны, и ϵ -версия, когда $\alpha_i = \alpha$, ϵ_i различны. Бифуркационная диаграмма для одного отображения (10) напоминает случай логистического отображения. Кодирование информации в сети очень простое, положительные значения $x > 0$ кодируют $+1$, а отрицательные -1 .

Идея распознавания образов при помощи решёток связанных отображений основана на том, что в решётке (9) при $\alpha_i = \alpha$ и $\epsilon_i = \epsilon$ существует область параметров (α, ϵ) , где она распадается на синхронизованные кластеры, отвечающие периодическому поведению. При этих параметрах система обладает довольно высокой степенью "сохранности информации", то есть поведение сильно зависит от начальных данных, и существует большая корреляция между $x(t)$ и $x(0)$. При больших значениях α или меньших ϵ в хаотическом состоянии это свойство теряется.

Идея методики состоит в том, чтобы определить локальный функционал энергии $E_i = -x_i \sum w_{ij} x_j$ и использовать хаос в качестве отжига чтобы разрушить нежелательные корреляции. Матрица связей w формируется согласно правилу Хебба из образов $\xi^{(k)}$, которые

надлежит запомнить, $w_{ij} = \sum_k \xi_i^{(k)} \xi_j^{(k)}$. Тогда в α -версии динамика x дополняется динамикой α :

$$\alpha_i(t+1) = \begin{cases} \alpha_i(t) + (\alpha_i(t) - \alpha_{\min}) \tanh(\beta E_i) & \text{every 16 steps,} \\ \alpha_i(t) & \text{otherwise.} \end{cases}$$

Значение α_{\min} отвечает фазе кластеров. Уменьшение α , согласно этому соотношению, вызывает переход к упорядоченной фазе когда локальная энергия становится достаточно малой.

Численные эксперименты показали, что эта система работает как ассоциативная память и её ёмкость составляет примерно $0.18N$. Похожие характеристики получены и для ϵ -версии алгоритма.

Другая версия алгоритма распознавания с переходом хаос \rightarrow порядок была предложена в работе [67]

Отметим, что описанная сеть не использует хаос для распознавания. Она использует его только в течение небольшого переходного периода, хотя поведение системы всегда остаётся нестационарным.

4.1.4 А что если переходный период — это навсегда или фильтр новизны

Всегда существует возможность, что при какой-либо комбинации входных параметров вместо кратковременного переходного хаоса может возникнуть хаотический аттрактор или длительность переходного периода будет так велика, что конца его можно не дожидаться. Скарда и Фриман [85] предположили, что такое состояние может означать ответ “я не знаю”, то есть то, что нейронная сеть столкнулась с чем-то таким, чему она не была обучена (см. также [55]). Подобное состояние, так же как и несоответствие прямого и обратного образов в сетях ART, в принципе может быть использовано как фильтр новизны, т.е. инициировать фазу обучения. Однако, насколько нам известно, примеры таких сетей опубликованы не были.

4.2 Хаос и поиск образов

Когда траектория движется по хаотическому аттрактору, она посещает его части одну за другой. Если связать различные части аттрактора с разными образами, траектория будет блуждать между ними. Вообще говоря, это блуждание могло бы быть использовано для организации ассоциаций: если траектория проводит длительной время вблизи лишь одного образа, а остальные посещает гораздо реже, то наиболее часто посещаемый образ можно считать “распознанным”, а если есть некоторая цепочка образов, которые посещает траектория, их можно рассматривать как “связанные” друг с другом. Заметим, что цепочку образов можно записать и в сеть типа Хопфилда. Быть может, однако, хаос способен менять эти комбинации, что давало бы возможность добавлять новые образы либо использовать один и тот же образ в нескольких цепочках. Подобные идеи так и не были полностью реализованы, но некоторые предварительные результаты получить удавалось.

Для исследования возможностей ассоциирования образов друг с другом Tsuda [93] предложил модель, которая в главных чертах напоминает сеть типа Хопфилда. Изначально образы запоминаются по правилу Хебба, но затем матрица связей динамически изменяет-

ся. Было показано, что ассоциирование образов друг с другом действительно имеет место, однако контролировать этот процесс в достаточной степени не удаётся.

Другой пример модели с ассоциативной хаотической динамикой был предложен в работе [1] и цитированной в ней литературе. Уравнения движения этой сети имели вид

$$x_i(t+1) = k_f x_i(t) + \sum_{j=1}^N w_{ij} f(x_j(t) + y_j(t))$$

$$y_i(t+1) = k_r y_i(t) - \alpha f(x_j(t) + y_j(t)) + a_i,$$

где $0 \leq x, y \leq 1$, $f(x) = 1/(1 + e^{-x/\epsilon})$, а образы запоминаются в соответствии с правилом Хебба, $w_{ij} = \sum_k (2\xi_i^{(k)} - 1)(2\xi_j^{(k)} - 1)$.

Результаты численного моделирования показывают, что траектории системы посещают окрестности запомненных образов. После огрубления (округления x до 0 или 1), некоторые состояния системы совпадают с запомненными образами. Однако сеть неспособна выделить только *один* образ, который можно было бы интерпретировать как выходной сигнал. Приведённые в упомянутой работе результаты демонстрируют некоторую зависимость поведения системы от начальных данных, однако не вполне ясно, соответствует ли эта разница (i) различным аттракторам, (ii) переходным процессам или же (iii) недостаточной длине траектории чтобы набрать хорошую статичтику.

Хаотическое блуждание среди запомненных образов изучалось иаже в [26] и было названо "хаотическое сканирование памяти".

Другая попытка использования хаоса в задачах распознавания была предпринята в [86, 87, 88, 89]. Было замечено, что образы можно распознавать при помощи наименьшего времени синхронизации, требующегося для полной или фазовой синхронизации входного образа с уже известными. Достоинством подхода является то, что данная процедура является достаточно общей и может быть использована как в хаотических, так и нехаотических сетях. Кроме того, конечное состояние не обязано быть устойчивой неподвижной точкой. Недостаток метода в том, что одновременно должно работать по крайней мере две копии динамической системы — для входного и для запомненного образа. Другой результат, который может быть полезен, состоит в том, что неустойчивые неподвижные точки можно превращать в устойчивые при помощи методик поиска корней уравнений. В принципе таким образом можно увеличивать ёмкость сети используя для хранения образов неустойчивые периодические орбиты. Однако для этого необходим способ отображать образы на эти орбиты и назад.

4.3 Хаос вместо неподвижной точки

Как уже говорилось в начале раздела 4, можно организовать многоаттракторные сети и сети с управляемым аттрактором не только при помощи аттракторов типа неподвижной точки, но также и с другими типами аттракторов. Один из простейших рецептов создания такой многоаттракторной сети описан в [10] — много идентичных маломодовых систем с периодическим или странным аттрактором объединяются в сеть типа Кохонена со взаимной конкуренцией, в которой "выживает" только одна из них; выжившая система

с ненулевой амплитудой служит индикатором распознанного образа. Для сетей с управляемым аттрактором такого простого и наглядного примера построено не было. Возможно, причина тому — сложная структура бифуркаций хаотических аттракторов, а также то, что различить аттракторы, отвечающие разным значениям параметров и интерпретировать их как разные отклики сети зачастую исключительно сложно. Тем не менее, одна из наиболее знаменитых хаотических сетей принадлежит именно к классу сетей с управляемым аттрактором. Это модель обонятельной луковицы, предложенная Фриманом.

В течение нескольких десятилетий исследования обонятельной системы были одной из основных целей У. Фримана и его коллег, см. [99, 31, 32, 30, 85] и ссылки в этих работах. После многих лет изучения строения обонятельной луковицы, они пришли к выводу, что только исследования нейронов и структуры их связей недостаточно для того, чтобы понять механизмы, ответственные за распознавание запахов. По этой причине они построили несколько математических моделей обработки информации обонятельной луковицей. Оказалось, что поведение моделей качественно согласуется с экспериментально полученными энцефалограммами, а динамика моделей хаотическая.

Модели весьма сложны. В наиболее известной из них каждая ячейка памяти описывается восемью дифференциальными уравнениями второго порядка, которые соответствуют группам нейронов различной специализации в пределах каждой ячейки. Все уравнения имеют один и тот же общий вид $\ddot{x}_i + A\dot{x}_i + Bx_i = G_i$, а члены в правой части отличаются для нейронов разных типов [99]. Некоторые из G_i включают входные данные X , у других есть члены с запаздыванием (зависящие от прошлых значений x). Есть нейроны, ответственные за связь с другими ячейками памяти, и для них G_i включает член вида $\sum K[i, j]Q(x[j])$, где Q — функция типа сигмоиды, а $x[j]$ соответствует такому же "связному" нейрону из j -й ячейки.

Информация в этой сети хранится в связях $K[i, j]$. Они могут принимать только два значения, K_{\min} и K_{\max} . Изначально все связи устанавливаются в K_{\min} , а чтобы записать образ, в котором ячейки i и j активны, соответствующая $K[i, j] = K[j, i]$ устанавливается в K_{\max} (правило Хебба).

Сеть работает следующим образом. В отсутствие внешних сигналов наблюдаются хаотические колебания на аттракторе системы. Когда предъявляется некоторый входной образ, система стабилизируется в каких-то областях бывшего аттрактора. Для пояснения основной идеи можно рассмотреть аттрактор с несколькими "крыльями", вроде аттрактора Лоренца. В состоянии базовой активности траектория посещает все части аттрактора, а внешний сигнал, "знакомый" системе, запирает траекторию на одном из крыльев (то есть новый аттрактор для данных значений параметров X съезживается до размеров одного крыла). Динамика остаётся хаотической, но только в новой меньшей области. То, что траектория не уходит с "крыла", можно декодировать в отклик сети, например, при помощи вычисления временных средних.

Существуют и другие хаотические модели обонятельной луковицы, например, [6], однако они не настолько хорошо согласуются с биологическими данными.

4.4 Рекуррентные сети как генераторы хаотических сигналов

Существует ряд работ, в которых нейронные сети рассматриваются вне контекста обработки информации, просто как модели некоторой биологической системы или как удоб-

ное представление уравнений движения динамической системы. При этом не изучаются вопросы вычислений, аппроксимации, ассоциативной памяти и т.п., см., например, [34, 2, 3, 61, 70] и некоторые другие работы. Мы не будем рассматривать этот тип сетей. В самом деле, совсем несложно сконструировать нейронную сеть в виде рекуррентного персептрона (Раздел 3), обладающую хаотическим поведением. Подобные сети не отвечают на вопрос о роли хаоса в обработке информации. Однако, если такая роль *существует*, подобные нейронные сети могут оказаться удобными генераторами хаоса [75].

4.5 Что же не так с хаотическими нейронными сетями?

Мы упомянули лишь некоторые из работ, связанные с хаотическими нейронными сетями, чтобы проиллюстрировать основные направления исследований в этой области. Однако общим для *всех* таких сетей, насколько нам известно, является то, что они не используются в практических приложениях. Единственное исключение — эксперимент, описанный в [100]. Многослойные персептроны, сети Кохонена или Гроссберга используются весьма широко, в то время как хаотические сети уже лет 15 остаются только объектом теоретических изысканий. Как гласит один из законов Мерфи, "если этим никто не пользуется, должна быть причина". В чём же причина? С нашей точки зрения, она в том, *как* используются нейронные сети. Существующий способ их использования можно назвать "изолированными вычислениями". Задача сети сводится лишь к тому, чтобы генерировать вполне определённый и всегда один и тот же отклик на заданный входной сигнал. Хаотическая динамика, которая неустойчива по определению, может лишь сделать подобные вычисления ненадёжными. Поэтому в такой схеме нет никакой естественной ниши для динамического хаоса.

В противоположность сетям такого типа, мозг всегда действует как часть тела. Он постоянно обрабатывает информацию, приходящую извне, и управляет телом с тем чтобы действовать и менять своё окружение. Иными словами, мозг функционирует как часть кольцевой связки: мозг — действие — окружающий мир — ощущение — мозг. Возможно, что именно *встроенность* мозга в тело позволит объяснить преимущества его работы в хаотическом режиме. Кроме того, хаос может просто возникнуть как результат положительных обратных связей в этом кольце.

Заметим, что проблема "встроенности интеллекта" широко обсуждалась в исследованиях по искусственному интеллекту в последние 15 лет, неплохой обзор приведён в [73]. Соответствующий подход, получивший название "behavior based robotics" или "embodied cognitive science" привёл к возникновению ряда эффективных практических решений и новых теоретических концепций. Более того, на примере небольшого робота, управляемого сравнительно несложной нейронной сетью, было показано, что замыкание связей через окружающий мир способно приводить к сложному, возможно, хаотическому поведению робота [73] (правда, насколько нам известно, никто не пытался количественно оценивать сложность такого поведения).

Другим естественным источником сложного временного поведения могут быть особые реализации нейронных сетей: использование ансамблей связанных осцилляторов для организации вычислений, например, [52, 48, 69]. В разделе 7 мы обсудим это более детально.

5 Замыкание связей в кольцо: хаос при комбинировании управляющей нейронной сети и управляемой системы

Как уже говорилось в разделах 3, 4, один из простейших способов получить хаос — это взять сеть-функцию, скажем, многослойный нерекуррентный персептрон, аппроксимирующий уравнения движения хаотической системы, и подать его выход обратно на вход. Получится хаотическая динамическая система. Однако она не занимается никакой полезной обработкой информации. Однако если мы поместим между выходом и входом сети некоторую систему, которой необходимо управлять, то обратная связь останется, а сеть будет решать задачу обработки информации. В данном разделе мы приведём довольно простые примеры того, как хаос может возникать в паре управляющая сеть – управляемая система. Хаотический сигнал может регистрироваться в любой части такой комбинированной системы. Этот эффект, кстати, может оказаться одним из источников хаотической активности мозга.

Сейчас мы рассмотрим только хаос, возникающий в задачах обработки информации. В следующем разделе мы рассмотрим роль хаоса в процессе обучения подобной системы контроллер-объект.

Рассмотрим динамическую систему

$$\dot{x} = \lambda x + f, \quad \lambda > 0. \quad (11)$$

При $f = 0$ у неё есть неустойчивая точка равновесия $x = 0$. Предположим, что мы можем управлять этой системой прикладывая "силу" f в дискретные моменты времени $t_k = \tau k$. После этого сила остаётся постоянной до следующего момента переключения t_{k+1} . Абсолютная величина силы $|f| = f_0$ остаётся постоянной, можно менять лишь её направление. Цель состоит в том, чтобы удерживать траекторию вблизи точки $x = 0$. Таким образом, в каждый из моментов t_k мы знаем $x(t_k)$, и нам следует принять решение относительно направления, в котором следует направить силу.

Эта задача очень проста. Обозначим $x_k = x(t_k)$. Поскольку $f_k = f(x_k)$ остаётся неизменной вплоть до t_{k+1} , из (11) следует, что $x_{k+1} = e^{\lambda\tau} x_k + (e^{\lambda\tau} - 1) \frac{f_k}{\lambda}$. Легко проверить, что выбор $f_k = -f_0 \text{sgn}(x)$ решает поставленную задачу и в результате поведение системы описывается следующим одномерным отображением,

$$x_{k+1} = g(x_k), \quad g(x) = \begin{cases} e^{\lambda\tau} x - A, & x \geq 0 \\ e^{\lambda\tau} x + A, & x < 0 \end{cases}, \quad A = \frac{(e^{\lambda\tau} - 1) f_0}{\lambda}. \quad (12)$$

График $g(x)$ приведён на Рис. 1. Легко видеть, что траектория остаётся вблизи неустойчивой точки при условии $|x(0)| < f_0/\lambda$. Так как $dg(x)/dx = e^{\lambda\tau} > 1$, возникает хаотический аттрактор с ляпуновским показателем, равным λ .

Управление системой может осуществлять "сеть" из одного порогового нейрона, который получает на входе x_k и генерирует сигнал ± 1 , показывающий направление силы. Поскольку аттрактор хаотический, последовательность сигналов нейрона будет выглядеть случайной. Источником этой случайности является дискретное управление неустойчивым состоянием равновесия.

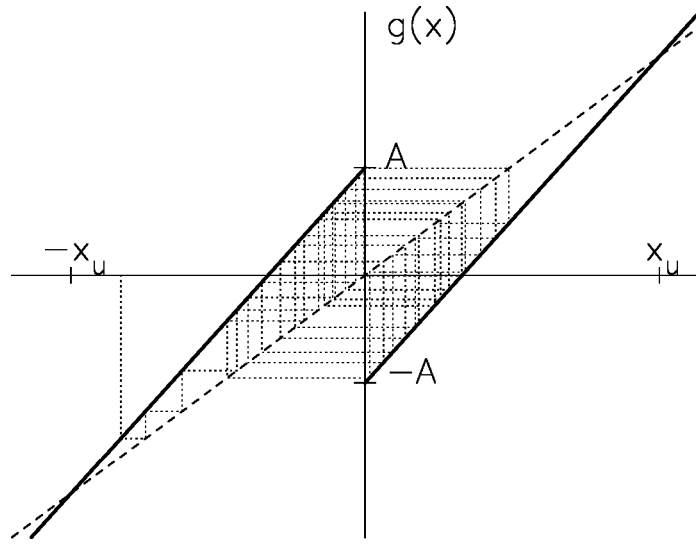


Рис. 1: Отображение, возникающее в результате дискретного управления для неустойчивой неподвижной точки и пример траектории.

Этот пример поясняет основную идею, но имеет два недостатка: (i) в нём нет обучения и (ii) нет настоящей нужды в использовании нейронной сети. Поэтому рассмотрим более сложные примеры дискретного управления. Множество таких примеров можно найти, например, в литературе по искусственному интеллекту [65].

Простейшим обобщением рассмотренного примера является перевёрнутый маятник, который необходимо удерживать в окрестности верхней точки. Можно, однако, показать, что эта проблема сводится к рассмотренному выше примеру, поскольку неустойчивое многообразие в задаче о перевёрнутом маятнике одномерно.

Более интересна задача о балансировании стержня на тележке, одна из хорошо известных тестовых задач в области "машинного обучения" [64, 12, 46]. Дана тележка, которая способна двигаться вдоль прямой от $-x_{\max}$ до x_{\max} . К ней одним концом прикреплён стержень таким образом, что он может вращаться в вертикальной плоскости, параллельной траектории тележки. Если стержень поставить вертикально, падая, он заставит двигаться тележку. Если же толкнуть тележку, можно тем самым влиять на динамику стержня. Состояние системы тележка-стержень характеризуется координатами x (положение тележки), \dot{x} (её скорость), θ (угол отклонения стержня от вертикали), и $\dot{\theta}$ (его угловая скорость), см. Рис. 2. Задача управления состоит в следующем. Через каждый промежуток времени длительностью τ контроллер получает значения переменных x , \dot{x} , θ , $\dot{\theta}$. Он может прикладывать к тележке силу $\pm f$ (например, запускать мотор), которая будет действовать в течение следующего τ -промежутка. Цель состоит в том, чтобы поддерживать величину угла θ в пределах $[-\theta_{\max}, \theta_{\max}]$, а положение тележки x в $[-x_{\max}, x_{\max}]$.

На первый взгляд, проблема эквивалентна управлению перевёрнутым маятником. Однако, если применить тот же самый алгоритм управления, окажется, что тележка довольно скоро уедет за допустимые пределы (Р. Хехт-Нильсен очень интересно описывает эксперименты с тележкой, которые проводились в его фирме [46]). Задача кажется простой, но при небольших размерах тележки требует характерного времени реакции около 0.1с,

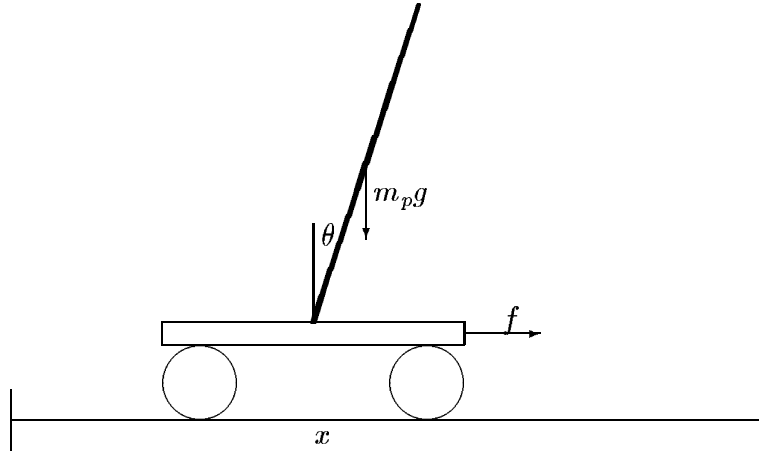


Рис. 2: Задача о балансировании стержня на тележке. Система управления должна после каждого интервала τ выбрать правильное направление f так чтобы угол отклонения стержня θ оставался в пределах $[-\theta_{\max}, \theta_{\max}]$, а тележка не уходила бы за границы дорожки, $-x_{\max} < x < x_{\max}$. В начальный момент тележка устанавливается в середину дорожки $x = 0$ а стержень — под некоторым углом θ_0 в допустимых пределах.

что превышает возможности человека; все с энтузиазмом берутся за дело, считая что уж у него-то получится, но...). А дело в том, что в этой задаче к одномерному неустойчивому многообразию стержня добавляется двумерное центральное многообразие тележки, так что траектория может свободно перемещаться уже в трёхмерном пространстве. Но в трёхмерном пространстве интуитивно угадать правильную стратегию уже не получается, нужно учиться. (Следует заметить, что с точки зрения теории управления эта задача довольно элементарная, но мы хотим, чтобы управляла тележкой обучающаяся нейросеть, которая теорию управления не изучала, а может действовать только путём проб и ошибок.) Детали возможного процесса обучения мы рассмотрим в следующем разделе. Здесь мы только отметим, что нейросетевой контроллер можно построить и обучить [77], а управление протекает в хаотическом режиме с одним положительным ляпуновским показателем. Мы записали активность нескольких нейронов управляющей сети и построили для них автокорреляционную функцию, Рис. 3. Их вид характерен для хаотических систем, хотя собственно архитектура использованной нейросети типа Кохонена никакой собственной хаотической динамики не допускает. Похожие результаты были получены и для другой модельной задачи — стабилизации неустойчивой химической реакции [77].

Таким образом, хаос может возникать в сложных петлях обратных связей, где нейронная сеть играет роль обучающейся системы управления. Можно, однако, спросить, а может ли хаос быть полезен для обработки информации или для обучения? Как показывают наши результаты, ответ утвердительный, но прежде чем его объяснить, необходимо описать основные идеи поощряемое обучения.

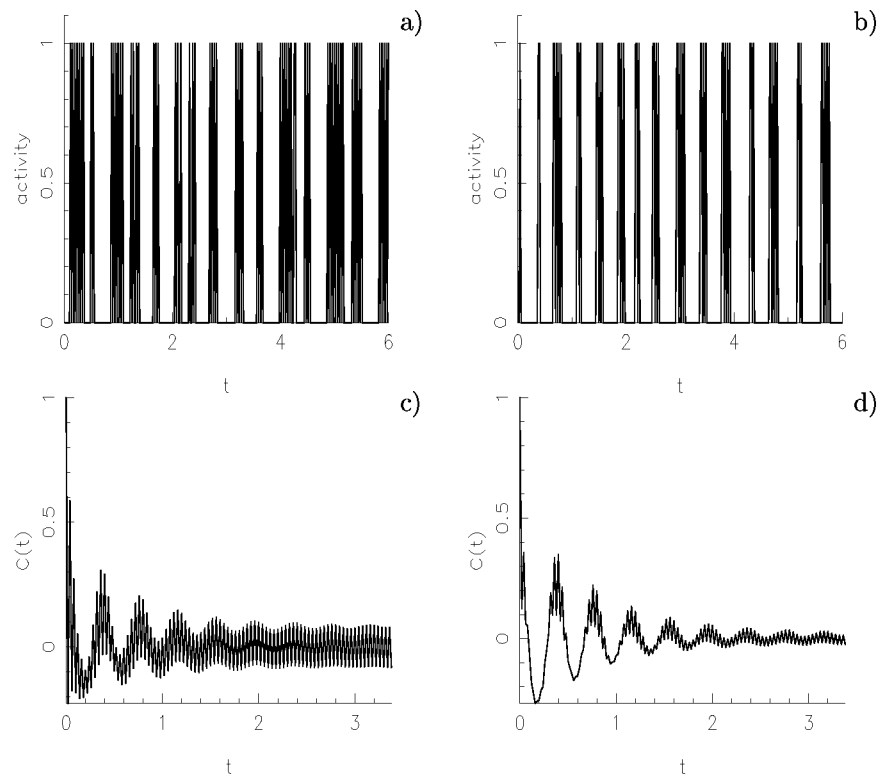


Рис. 3: “Энцефалограмма” для нейронной сети, управляющей тележкой со стержнем. Графики а и б демонстрируют временную активность двух нейронов (1 когда нейрон активен и 0 в противном случае). На графиках с и д показана соответствующая автокорреляционная функция. Видны следы как детерминизма, так и хаотичности.

6 Хаос и обучение поощрением (reinforcement learning)

6.1 Что такое обучение поощрением?

В большинстве книг по нейронным сетям рассматривается два типа обучения, "с учителем" (supervised) и "без учителя" (unsupervised). Если для каждого обучающего входного сигнала X известен правильный ответ Y , и этот ответ используется для подстройки связей сети, то говорят, что обучение идёт с учителем. Оно типично для большинства нейронных сетей. Если же обучение идёт совсем без корректирующего сигнала, то говорят, что обучение идёт без учителя. Примерами могут служить рассмотренные сети Кохонена и Гроссберга (ART), когда для входных сигналов заранее неизвестно, какому кластеру они должны принадлежать. Однако возможна и промежуточная ситуация, когда правильный отклик неизвестен, однако некоторая оценка работы сети может быть дана. Такая оценка обычно имеет вид скалярного "поощряющего" сигнала r : в случае успеха $r > 0$, при неудаче $r < 0$, а в нейтральных случаях $r = 0$. Соответствующий тип обучения был назван обучение поощрением (reinforcement learning) или "обучение с критиком". В традиционных нейронных сетях он практически не используется. Однако он оказывается весьма полезен в ситуациях, где известна конечная цель, но неизвестен правильный способ её достижения (можно поощрять приближение к цели или наказывать за удаление от неё или за медлительность — как, цель ещё не достигнута?!). Примером может служить задача о стержне на тележке, когда трудно сказать, в какую сторону сейчас надо было толкать тележку, в то время как заметить выход параметров за допустимые пределы несложно. В этой связи следует заметить, что поощрение может быть запаздывающим. Примером может служить партия в шахматы: поощрение в виде выигрыша, проигрыша или ничьей получается лишь в результате большого числа ходов, и сказать, какой из них был правильным, а какой — нет, зачастую очень сложно. Проблема определения ответственности каждого действия за конечный результат широко обсуждалась еще в начале 60-х годов и получила название "credit assignment problem".

Весьма вероятно, что высокоорганизованные живые организмы используют какой-либо тип обучения поощрением. У них есть довольно сложный источник поощряющего сигнала, такой как чувство боли или удовлетворения. Видимо, существование этой системы даёт этим организмам преимущество. Похожие источники поощряющего сигнала использовались и для мобильных роботов [73], они были названы оценочной подсистемой (value system).

Ввиду бурного развития систем искусственного интеллекта и автономных роботов, следует ожидать и всё возрастающего использования методов обучения поощрением. Проблема, однако, заключается в том, что в настоящее время нет достаточно общей теории таких методов. Сама концепция обучения поощрением недостаточно конкретна и может относиться к слишком разным системам и ситуациям. Однако весьма большой прогресс был достигнут в одном из классов задач, известных как марковский процесс принятия решений (Markov decision process, MDP) или динамическое программирование (Dynamical Programming, DP).

Формальное описание MDP включает (i) *окружение* (environment), характеризуемое состоянием s , (например, система, которой необходимо управлять, для тележки со стержнем $s = \{x, \dot{x}, \theta, \dot{\theta}\}$), и (ii) *агент* (управляющая система), который может выполнять действие

a (в примере с тележкой действием было приложение силы $\pm f$). Для простоты будем считать, что s и a могут принимать только дискретные значения. Агент получает информацию о состоянии s_t в момент t , и предпринимает действие a_t , которое влияет на состояние окружения. После каждого действия окружение меняет своё состояние и агент получает информацию о следующем состоянии s_{t+1} и скалярный поощряющий сигнал r_{t+1} . Правило $\pi(s, a)$, по которому агент выбирает свои действия, называется *политикой* (policy). Обычно $\pi(s, a)$ — это вероятность выбрать действие a в состоянии s . Детерминированный выбор означает, что только для одного из действий вероятность ненулевая.

Задача агента состоит в том, чтобы выработать *оптимальную политику*, которая даёт максимальное интегральное поощрение за большое или бесконечное число действий. В последнем случае сумма $\sum_{k=1}^{\infty} r_k$ может оказаться бесконечной и используют взвешенное (discounted) интегральное поощрение $\sum_{k=1}^{\infty} \gamma^{k-1} r_k$, $0 < \gamma < 1$ (обычно величина $|r_k|$ ограничена).

Вплоть до настоящего момента ситуация выглядела весьма общей. Ограничивающим предположением, сильно облегчающим теоретическое исследование, является то, что динамика окружения предполагается марковской. Обозначим через $P = P(a, s, s')$ вероятность перехода окружения из s в s' после действия a , а соответствующее поощрение $R = R(a, s, s')$ (r_k всегда равно одному из R с соответствующими аргументами). Предполагается, что вероятности P и поощрения R не зависят от времени, от предыдущих состояний окружения и предшествующих действий агента. Следовательно, динамику окружения можно описать в терминах управляемой цепи Маркова. Далее мы будем полагать её эргодической.

Если все P и R известны, можно вычислить среднее взвешенное поощрение для состояния s , действия a и избранной политики π путём усреднения по всем возможным будущим траекториям вдоль марковской цепи,

$$Q^\pi(s, a) = \sum_{s'} P(a, s, s') \left(R(a, s, s') + \gamma \sum_{a'} \pi(s', a') \sum_{s''} P(a', s', s'') (R(a', s', s'') + \dots) \right).$$

Величины $Q^\pi(s, a)$ были названы *ценность действия* (action values). Поскольку мы предположили, что π , P и R не зависят от времени, в правой части возникают те же ценности действий Q^π , поэтому

$$Q^\pi(s, a) = \sum_{s'} P(a, s, s') \left(R(a, s, s') + \gamma \sum_{a'} \pi(s', a') Q^\pi(s', a') \right). \quad (13)$$

Это линейная система уравнений для Q^π , которую можно решить стандартными методами.

Почему ценности действий Q^π так важны? Дело в том, что с их помощью можно улучшать существующую политику, до тех пор пока не будет найдена оптимальная политика, приносящая максимальный выигрыш. После того как все Q^π найдены, естественно изменить текущую политику π так, чтобы в каждом состоянии выбирать наиболее ценное действие $a = \arg \max_{a'} Q^\pi(s, a')$. В результате получается новая политика π_1 . Как показал более 40 лет назад Р. Беллман [14], для новой политики $Q^{\pi_1}(a, s) \geq Q^\pi(a, s)$. Затем можно проделать то же самое с новыми ценностями действий $Q^{\pi_1}(a, s)$ и получить следующую политику π_2 и так далее. Этот процесс итерирования политики сходится к оптимальной

политике π_* , которой соответствуют $Q^*(s, a)$. Оптимальных политик может быть и несколько, но ценности действий у всех одни и те же. Существует весьма эффективный численный алгоритм для итерирования политики [14, 83].

Существующие методы обучения поощрением (Reinforcement Learning, RL) [83, 57] развиты для случая, когда все предположения динамического программирования справедливы, но переходные вероятности P и поощрения R неизвестны. Это означает, что нет возможности найти оптимальную политику при помощи уравнений (13). Проблема напоминает управление тележкой со стержнем, можно действовать только методом проб и ошибок. Как же нам научиться удерживать стержень?

Идея методов обучения поощрением состоит в том, чтобы оценить те же самые ценности действий Q , но путём усреднения по времени, а не по ансамблю (13), поэтому нам нужна эргодичность. Когда агент в момент t выбирает действие a_t в состоянии s_t и на следующем шаге получает поощрение r_{t+1} , последнее может быть использовано для коррекции оценки $Q(s_t, a_t)$. Если сделать достаточно много попыток, оценки могут стать достаточно точными и быть использованы для улучшения политики. Хотя как правило улучшение текущей политики не требует точного знания всех Q , необходимо только знать, для какого действия a величина $Q(s, a)$ является наибольшей. А эта информация часто может быть получена за разумное время.

Существующие методы обучения поощрением обычно подразделяют на два больших класса [83, 57]: Методы Монте-Карло, когда Q оценивают путём непосредственного усреднения поощрений, полученных после данной комбинации состояние-действие, и методы временных разностей (Temporal Differences, TD), когда оценки получают при помощи итерационных схем. Методы Монте-Карло требуют очень много повторений и сходятся медленно. Мы не будем их здесь рассматривать. Согласно [83, 57], TD-методы более популярны и обычно сходятся гораздо быстрее.

TD методы основаны на следующем несложном соотношении: когда мы пытаемся оценивать Q экспериментально, оценки связаны как $Q(s_t, a_t) = \sum_{k=0}^{\infty} \gamma^k r_{k+t+1} = r_{t+1} + \gamma Q(s_{t+1}, a_{t+1})$, а потому $\Delta_t = r_{t+1} + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t) = 0$. Если $\Delta_t \neq 0$, это означает, что текущая оценка нуждается в коррекции. Методика выглядит следующим образом. Используются оценки ценностей действия, зависящие от времени, $Q(s, a, t)$. Начальные значения $Q(s, a, 0)$ можно выбирать достаточно произвольно. Затем следует выбрать некоторую политику π и после каждого шага итерационно корректировать ценности действий:

$$Q(s, a, t + 1) = Q(s, a, t) + \alpha_t e(s, a, t) \Delta_t. \quad (14)$$

Это соотношение содержит два дополнительных фактора — скорость обучения α_t и так называемые коэффициенты пластичности (eligibility traces) $e(s, a, t)$, о которых мы поговорим чуть позже. Для расчёта корректирующего члена Δ чаще всего используют две основные методики.

а) "Sarsa" (в качестве имени используется последовательность $s_t, a_t, r_{t+1}, s_{t+1}, a_{t+1}$, применяемая для расчёта Δ)

$$\Delta_t = r_{t+1} + \gamma Q(s_{t+1}, a_{t+1}, t) - Q(s_t, a_t, t).$$

Метод даёт оценки Q для текущей политики. Тем не менее, если в свою очередь текущие значения Q используются для формирования политики, например, каждый раз выбирается

действие с наибольшей $Q(s_t, a, t)$ (обычно выбор сложнее, но об этом тоже позднее), то оказывается, что методика допускает одновременно и улучшение политики.

b) "Q-learning",

$$\Delta_t = r_{t+1} + \gamma \max_a Q(s_{t+1}, a, t) - Q(s_t, a_t, t).$$

Метод сходится к значениям Q^* для оптимальной политики π^* почти независимо от текущей политики π . Зная Q^* , нетрудно получить и саму оптимальную политику π^* .

В отношении выбора скорости обучения α серьёзных теоретических ограничений нет. Доказательство сходимости методов RL [83, 57] требует, чтобы α_t убывали со временем так чтобы $\sum_t \alpha_t = \infty$, $\sum_t \alpha_t^2 < \infty$ (требования методов стохастической аппроксимации), например, $\alpha_t = \alpha_0/t$. Иногда описанные методы хорошо работают и для неубывающих α_t , которые к тому же могут даже не быть очень маленькими. Для некоторых задач известно, что наилучший выбор $\alpha = 1$ [57].

Теперь обратимся к коэффициентам пластичности. Изначально они появились в моделях обучения нейронов для того чтобы воспроизвести эффекты сходные с условными рефлексам [84]. Такие рефлексы связывают два события, "нейтральное" и "существенное", разделённые временным промежутком. Идея состояла в том, что нейроны и связи, ответственные за обработку нейтральных событий, в течение некоторого времени после них сохраняют возможность изменяться (пластичны), поскольку сохраняют память о недавнем событии. Если в течение этого промежутка произойдёт существенное событие, оно может изменить связи, сохранившие пластичность. В результате после нескольких повторений нейтральное событие получает способность вызывать такой же эффект, как и существенное. Соответствующая математическая модель нейронов, помимо обычных связей w_{ijt} между ними, включала также величины e_{ijt} , названные "eligibility traces", по-видимому, наиболее адекватный перевод — коэффициенты пластичности. Правило изменения весов имело вид $\Delta w_{ijt} \sim e_{ijt} g_t$, где g_t — сигнал, порождённый "существенным" событием. Первоначально все $e_{ij0} = 0$. Когда происходит нейтральное событие и связи обретают пластичность, соответствующий коэффициент e становится равным 1 (или увеличивается на 1). Затем он экспоненциально затухает, $e_{t+1} = \lambda e_t$, $\lambda < 1$. Таким образом, в течение некоторого времени после установки e , соответствующая связь может меняться, однако в дальнейшем эта способность пропадает до новой установки.

В обучении поощрением эта идея была использована как попытка решить упоминавшуюся проблему credit assignment. Когда корректирующий член Δ вычислен, было бы желательно скорректировать Q не только для последнего состояния s_t , но и для предшествующих состояний, поскольку последний сигнал поощрения частично характеризует и предыдущие действия. Чтобы это сделать, следует присвоить этим состояниям коэффициент ответственности. Рецепта, как этого добиться в общем случае, нет, но для эргодических цепей Маркова эффект каждого действия или пребывания в некотором состоянии должен экспоненциально убывать. Следовательно, по крайней мере для некоторых задач, экспоненциально убывающие коэффициенты пластичности могли бы частично решить проблему credit assignment. Соответствующий класс алгоритмов RL был назван TD(λ), где λ обозначает скорость затухания коэффициентов пластичности. Для каждой пары состояние-действие вводится переменная $e(s, a, t)$, $e(s, a, 0) = 0$. Как только окружение оказывается в состоянии s_t и выбирается действие a_t , соответствующая $Q(s_t, a_t)$ получает способность меняться и мы присваиваем $e(s_t, a_t, t) = 1$ (в некоторых версиях $e(s_t, a_t, t) = e(s_t, a_t, t) + 1$).

Для всех прочих пар состояние-действие коэффициенты e экспоненциально затухают, так что окончательно

$$e(s, a, t) = \begin{cases} 1, & (s, a) = (s_t, a_t), \\ \lambda e(s, a, t - 1), & (s, a) \neq (s_t, a_t). \end{cases}$$

Обучение может быть разбито на эпизоды. Например, для стержня на тележке после падения стержня или ухода тележки за допустимые пределы тележка возвращается в центр дорожки, а стержень снова устанавливается почти вертикально. В таких случаях коэффициенты пластичности также следует установить в ноль.

Согласно [83, 57], коэффициенты пластичности могут существенно ускорить сходимость алгоритмов RL, особенно в ситуациях с запаздывающим поощрением, когда ненулевое r появляется лишь в результате многих действий. В отношении скорости затухания λ , так же как и весового множителя γ , единственным теоретическим ограничением является $0 \leq \lambda < 1$, $0 < \gamma \leq 1$. Некоторые практические рекомендации по выбору α , λ и γ можно найти в статье Дж. Тесауро [90] (автора получившей известность программы TD-gammon, играющей в нарды — backgammon — на уровне мировых мастеров; в основе программы лежат методы обучения поощрением, а для аппроксимации ценности состояний применяется трёхслойный перцептрон с несколькими десятками скрытых нейронов). Однако качество работы методов RL может существенно зависеть от значений параметров, поэтому в каждом конкретном случае может оказаться необходимым проведение специальных тестов. Приложения обучения поощрением в настоящее время довольно многочисленны, помимо упомянутой игры в нарды это управление сложными механическими системами, роботами, лифтами в многоэтажных зданиях, контроль хаоса и др., см., например, [83, 57, 92, 36, 97].

Какое отношение всё вышеизложенное имеет к нейросетям? Связи с нейросетями многочисленны. Во-первых, методы обучения поощрением используют функции, аппроксимация которых необходима в некоторых вариантах методов [57, 83]: $\pi : s \rightarrow a$, $Q : s \times a \rightarrow R^1$, $R, P : s \times s \times a \rightarrow R^1$, иногда и некоторые другие. Подобные аппроксимации могут быть получены с помощью какой-либо из аппроксимирующих сетей, например, многослойного перцептрона. Во-вторых, состояние s часто изменяется непрерывно. Версии динамического программирования и обучения поощрением для непрерывного случая также существуют, но они гораздо сложнее, чем в дискретном случае. Поэтому часто пространство возможных состояний разбивается на множество областей, каждая из которых рассматривается как одно дискретное состояние. Эту процедуру квантования состояний естественно поручить нейросети для векторного квантования, такой как сеть Кохонена или ART [18, 59]. Например, нейросетевой контроллер может иметь вид двухступенчатой сети Кохонена. В качестве входного сигнала сеть получает s , причём каждый нейрон отвечает одному дискретному состоянию. После выбора победителя, нейрон, отвечающий текущему состоянию, включается и посылает сигнал следующей сети через ряд выходных связей с весами, равными Q . Во второй конкурентной сети каждый нейрон соответствует одному действию. После соревнования выбирается действие с наибольшим Q . Таким образом, нейронные сети — удобный инструмент методов RL, иногда даже используется термин "нейродинамическое программирование" [15] (не путать с нейролингвистическим программированием из области психологии).

Существует и связь со сложной динамикой и хаосом. Во-первых, как говорилось в предыдущем разделе, нейросетевой контроллер с поощрительным обучением объединённый с

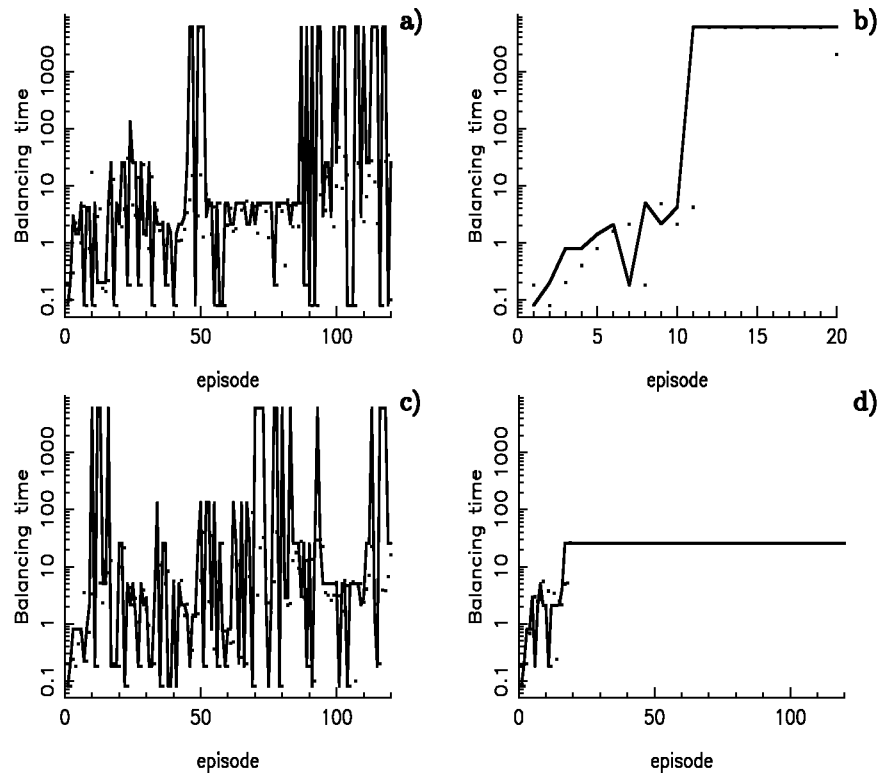


Рис. 4: Примеры обучения поощрением (Q -learning) для тележки со стержнем. Обучение разбивается на эпизоды, эпизод заканчивается когда x или θ выходят за допустимые пределы (контроллер в этом случае получает отрицательный поощряющий сигнал) или управление успешно протекает вплоть до момента $T = 6000$. Последнее означает, что контроллер способен удерживать стержень очень долго, скорее всего, бесконечно долго. Продолжительность каждого эпизода показана точками. Иногда использовалась ϵ -жадная политика (графики а и с), и стержень иногда падал только из-за случайного действия. После каждого эпизода тестировалась получающаяся жадная политика, продолжительность удержания стержня для которой показана сплошной линией. Графики соответствуют различным параметрам Q -learning. При хорошем выборе (b) обучение весьма эффективно. Из графиков а и с видно, что при некоторых параметрах возможны колебания между различными политиками: контроллер находит хорошую политику, потом забывает, потом находит снова.

неустойчивой динамической системой, может научиться поддерживать её в хаотическом состоянии. На Рис. 4 показан пример такого обучения для системы тележка-стержень. Во-вторых, хаос может быть необходим контроллеру для получения хорошей политики при помощи методов RL с освоением новых действий (exploration). Рассмотрим этот вопрос более детально.

6.2 Обучение поощрением, освоение действий и хаос

Обсудим теперь вопросы формирования политики, т.е. согласно каким же критериям следует выбирать действия в каждом заданном состоянии. В задачах динамического программирования, когда информация о свойствах окружения полна, ценности действия Q^* для оптимальной политики могут быть рассчитаны, оптимальным является способ выбора действий с наибольшим Q^* . Однако при обучении методом проб и ошибок такая политика часто оказывается далеко не лучшей, поскольку известны не точные значения Q^* , а лишь их оценки. Даже если часть значений Q известна точно, всегда есть вероятность того, что лучшее действие ещё не было испробовано. Это в свою очередь значит, что истинные Q^* ещё неизвестны. Если агент использует только наилучшие из известных действий, он может никогда не найти оптимальной политики. Следовательно, иногда необходимо пробовать неоптимальные действия. Агент должен осваивать новые возможности путём проб и ошибок. С другой стороны, если агент предпринимает слишком много плохих действий, его эффективность снижается. Поэтому возникает *дилемма освоения и эксплуатации* [83, 91]. Когда мы рассказывали про метод Q -learning, мы говорили, что он даёт значения Q^* для оптимальной политики *почти* независимо от текущей политики π : текущая политика должна давать возможность осваивать лучшие действия.

Когда агент всегда выбирает действие с максимальной оценкой Q , такая политика называется *жадной* (greedy). Было показано, что следование чисто жадной политике без освоения может привести в результате к неоптимальной политике [83]. С другой стороны, иногда следование жадной политике вполне оправдано. Необходимый элемент освоения часто может возникнуть сам собой благодаря блужданию в пространстве политик во время сходимости оценок Q . Можно дополнительно стимулировать освоение, устанавливая высокие начальные приближения для Q (“оптимистичные” начальные данные) и побуждая агента попробовать большинство действий во время обучения.

Для того, чтобы гарантировать освоение новых действий, часто используют политику, называемую *ϵ -жадной*. Допустим, что в состоянии s есть n возможных действий. Тогда мы выбираем лучшее действие с вероятностью $1 - \epsilon$, а любое иное действие с вероятностью $\epsilon / (n - 1)$. Когда оценки Q почти сойдутся, освоение уже будет только портить результаты из-за не лучших действий. Поэтому рекомендуется постепенно уменьшать ϵ с течением времени [83].

Предположим теперь, что окружение нестационарно, хотя изменения редки или невелики, так что модель марковской цепи по-прежнему можно использовать на временах, достаточных для сходимости Q . Благодаря нестационарности, оптимальная политика со временем может измениться. В этом случае жадная политика может оказаться опасной: освоение благодаря переходным процессам и начальным данным теперь отсутствует, и новую оптимальную политику можно никогда не найти. В таких обстоятельствах важность могут приобретать специальные меры по освоению новых действий, например, в

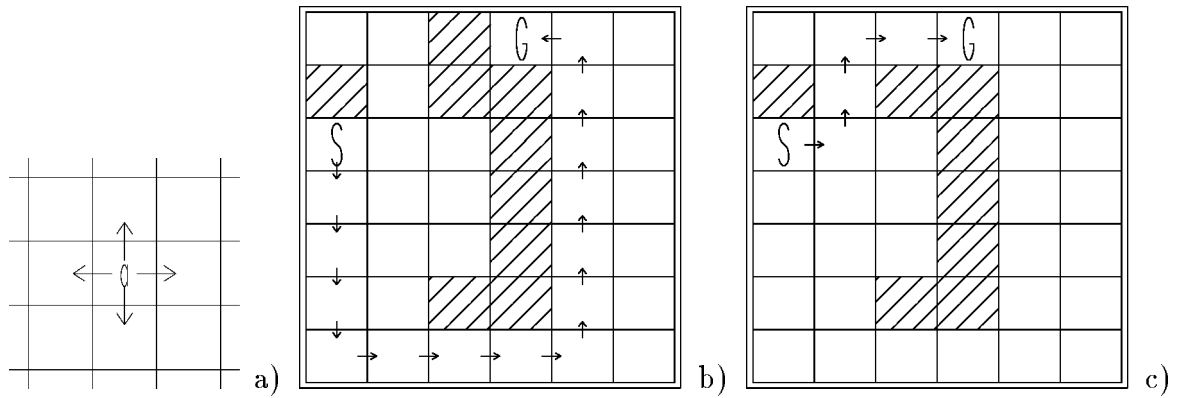


Рис. 5: Тестовая задача для хаотической политики — движение по решётке. (а) В каждом состоянии агент может выполнять 4 действия: вверх, вниз, вправо, влево, если в выбранном направлении нет пустой клетки, агент остаётся на месте. (б) Начальное окружение, агент стартует с клетки “S” и должен достичь цели “G”. Заштрихованные клетки агенту недоступны и кратчайший путь занимает 15 шагов. На каждом шаге агент получает поощрение $r = -1$, пока не достигнет цели; тогда он получает $r = +1$ и возвращается на стартовую клетку. (с) В момент $t = 2010$ одна из заблокированных клеток открывается и появляется короткий путь из 5 шагов. Чтобы открыть этот новый путь, агент должен использовать политику с освоением действий.

виде ϵ -жадной политики. Это можно продемонстрировать на примере движения на решетке — типичной тестовой задаче методов обучения поощрением [57, 83].

В задаче навигации на решётке окружение состоит из клеток, а агент может двигаться из данной клетки в одну из соседних с ней, Рис. 5а. Возможными действиями являются вверх, вниз, влево и вправо. Если в выбранном направлении нет клетки или она заблокирована (заштрихованные клетки на Рис. 5), то агент остаётся на прежнем месте. Движение начинается из клетки, отмеченной “S”, агент получает наказание (отрицательное поощрение) $r = -1$ после каждого действия, пока он не достигнет своей цели — клетки “G”. Когда цель достигнута, агент получает поощрение $r = 1$ и мгновенно отправляется снова на стартовую клетку. Таким образом, задача агента — найти кратчайший путь из “S” в “G” с тем, чтобы минимизировать наказание. Количество состояний s в этой задаче равно числу пустых клеток минус 1 (“G”), и в каждом состоянии возможны 4 действия. События, происходящие в течение одного путешествия от “S” до “G” образуют *эпизод*. Продолжительность текущего эпизода T можно рассматривать как меру качества текущей политики. Мы будем характеризовать текущую ϵ -жадную политику усреднённым отношением минимально возможной продолжительности T_{\min} к T : $q = \langle T_{\min}/T \rangle$, осреднение делалось по 100 различным прогонам для одного и того же ϵ . При выборе действий мы присваивали им номера, и, как оказалось, иногда результаты зависели от порядка нумерации. Поэтому на графиках результаты будут приведены для всех 24 возможных способов перенумеровать 4 действия.

Если в этой задаче окружение остаётся неизменным, специальных способов освоения действий не требуется, оба метода, *sarsa* и *Q-learning* с жадной политикой находят оптимальную или почти оптимальную политику. Дополнительное освоение может лишь ис-

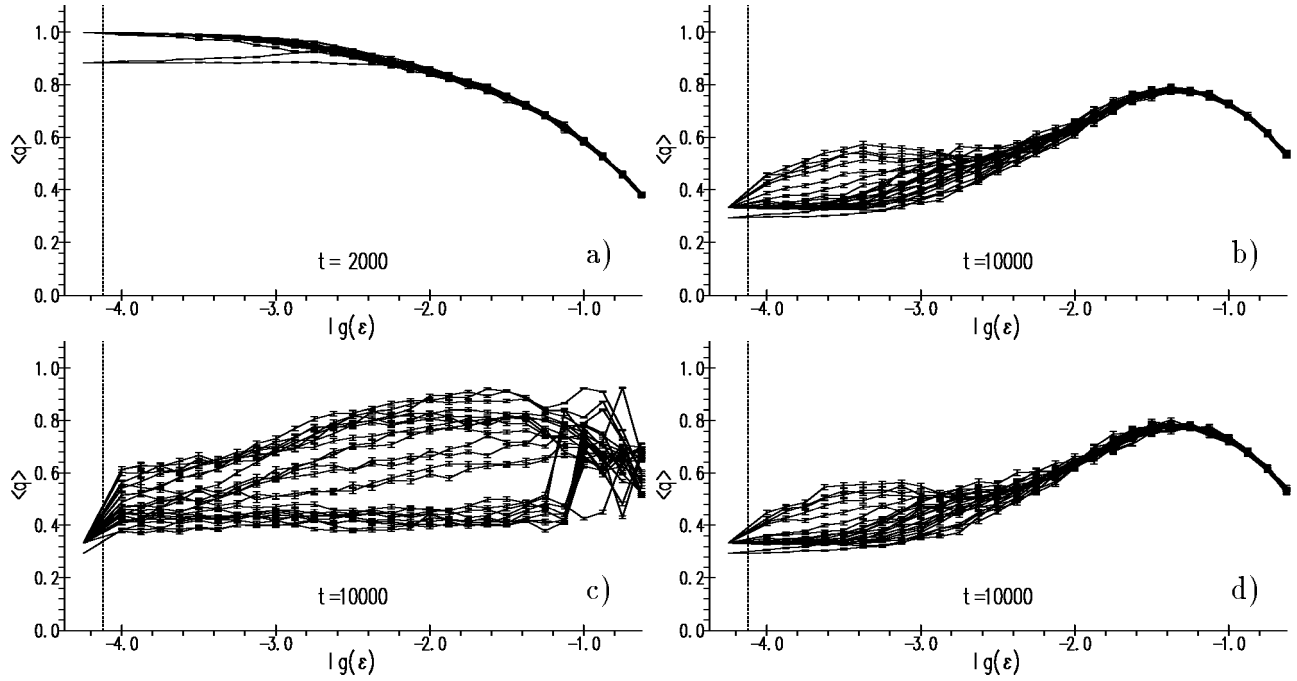


Рис. 6: Результаты для тестовой задачи на Рис. 5. Разные кривые на графике соответствуют разным нумерациям действий. Риски отмечаю оценку погрешности после усреднения по 1000 запусков. Самая левая точка, отделённая вертикальным пунктиром, соответствует $\epsilon = 0$. (a) Для стационарного окружения освоение только портит эффективность: чем больше ϵ , тем меньше q . (b) После изменения окружения освоение может повысить эффективность: существует оптимальный уровень шума, для которого эффективность наибольшая. (c) Нехаотическое освоение, ждя выбора действий используется траектория логистического отображения при a_∞ , регулярность ведёт к сильной зависимости результата от нумерации действий. (d) Хаотическое освоение, каждая 8-я точка для нейрона с надёжным (robust) хаосом. Результаты практически те же, что и для случайного освоения. Следовательно, хаос можно использовать, чтобы учиться новым действиям.

портить результаты, Рис. 6а, поэтому мы использовали нестационарное окружение [76]. После 2010 шага, окружение меняется: самая верхняя из заблокированных клеток на Рис. 5b открывается, Рис. 5с, и в новом окружении кратчайший путь требует уже только 5 шагов вместо 15. Без специального освоения приспособиться к новому окружению невозможно. Теперь уже существует оптимальная степень освоения, графики на Рис. 6b имеют чёткий максимум.

Освоение требует случайности в алгоритме, то есть обучающаяся система должна иметь источник случайного сигнала. И здесь мы подходим к вопросу о *естественном месте динамического хаоса в процессе обучения*. Хаос может служить источником псевдослучайных решений при использовании политики с освоением действий. Для проверки этой гипотезы мы повторили предшествующий эксперимент со случайной, регулярной и хаотической политиками. При выборе действий ϵ -жадной политики мы использовали последовательность псевдослучайных чисел, а также числа, порождённые нехаотической и хаотической динамическими системами, Рис. 6с,d. Для хаотической последовательности, где не наблюдаются существенные корреляции (используется каждое k -е число или данные от k независимых систем) результаты практически такие же, как и для случайных данных.

Среди хаотических систем наилучшие результаты были получены для нейрона с "надёжным хаосом" (robust chaos), т.е. отображения $x_{t+1} = |\tanh s(x_t - c)|$ [11, 75]. Следовательно, нейронные сети могут как использоваться для реализации методов обучения с поощрением, так и служить источником инициативы для освоения действий.

7 Гамильтоновы нейронные сети: распознавание образов без диссипации

В последние годы активно развиваются методы квантовых вычислений и квантовой обработки информации, см., например, обзоры [56, 68, 98, 94]. Возможно, что мозг также использует в своей работе квантовые механизмы [71]. В настоящее время трудно сказать, какие именно квантовые устройства появятся в будущем. Однако одно остаётся бесспорным, квантовые компьютеры были и будут гамильтоновыми системами. Важность соответствий между квантовыми и классическими системами была осознана еще в первые годы развития квантовой механики. Хотя построение подобных соответствий часто сталкивается с большими трудностями, борьба за лучшее понимание таких соответствий будет оставаться необходимой. В этой связи необходимо отметить, что ни одна из описанных выше нейронных сетей не может иметь квантовых аналогов по той простой причине, что они не являются гамильтоновыми системами. Это заставило нас обратиться к гамильтоновым нейронным сетям.

Гамильтоновы нейросети (ГНС) могут быть линейными или нелинейными, хаотическими или регулярными, автономными или неавтономными. Возможностей много. Простейшая — это линейная автономная гамильтонова система. Надо заметить, что эта система будет консервативной, в отличие от всех ранее рассмотренных нейронных сетей, которые являются диссипативными. В традиционных нейронных сетях диссипация играет очень важную роль. Она служит фильтром для удаления ненужной (шумовой) информации из входных данных при решении задачи распознавания образов. Само существование ат-

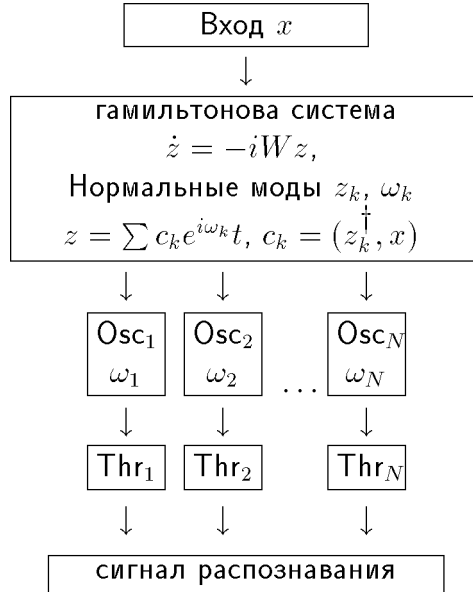


Рис. 7: Общая схема гамильтоновой нейросети. Вход сети x используется как начальные данные для линейной гамильтоновой системы. Последняя раскладывает их по нормальным модам z_k . Моды возбуждают осцилляторы Osc с частотами ω_k , равными частотам мод. Благодаря резонансу, амплитуда осциллятора растёт пропорционально амплитуде соответствующей моды. Пороговый детектор Thr определяет, для какого осциллятора амплитуда первой превысила порога, и формирует сигнал распознавания. Обучение системы заключается в формировании матрицы W .

тракторов также основано на диссипации. Следовательно, для распознавания образов при помощи ГНС необходимы методики, отличные от традиционных.

В данном разделе мы хотим изложить некоторые идеи, которые могли бы быть использованы при построении гамильтоновых нейронных сетей. Мы приведём описание метода проекций на подпространство и конкурентную архитектуру. Совместное их использование позволяет построить гамильтонову нейросеть, способную распознавать образы.

Линейную гамильтонову динамическую систему можно записать в виде $\dot{z} = (U + iW)z$, где U антисимметричная, а W симметричная действительные матрицы. Здесь $z = x + ip$ — N -мерный комплексный вектор. Положим для простоты $U = 0$. Тогда эта система эквивалентна N осцилляторам с частотами ω_k — собственными значениями W . Каждой колебательной моде отвечает собственные вектор e_k матрицы W . Эти моды называют нормальными. Когда мы вводим начальные данные, они раскладываются по этому набору мод, $z(t) = \sum_k (z_0, e_k) e^{i\omega_k t}$, причём амплитуда каждой моды пропорциональна проекции входного образа на e_k .

Если бы мы могли построить матрицу W таким образом, что e_k были бы запомненными образами, то амплитуды мод отвечали бы степени близости входных данных к каждому из запомненных образов. Следовательно, мода с наибольшей амплитудой (наиболее близкая) может рассматриваться как распознанный образ.

Отбор мод может быть выполнен с использованием эффекта резонанса для осцилляторов с возбуждением. Возьмём N осцилляторов с частотами ω_k , равными частотам нормальных мод, и будем возбуждать их сигналом от гамильтоновой системы. Поскольку все осцилляторы находятся в резонансе, их амплитуда будет расти, и наибольшая скорость роста будет у осциллятора, соответствующего наиболее возбуждённой моде. Следовательно, необходимо определить, для какого осциллятора амплитуда первой преодолет некоторый порог. Это событие отмечает нужный осциллятор и потому способно порождать сигнал распознавания. Заметим, что удобным видом осциллятора могла бы быть система, у которой потенциал имеет две ямы. Начальные данные для осциллятора соответствуют дну одной из ям, а её присутствие во второй яме генерирует сигнал распознавания (Рис. 7).

Единственная проблема — это найти способ записать набор вообще говоря неортогональных образов в ортогональные нормальные моды не искажая их формы. Эту проблему можно решить при помощи скрытых размерностей.

Предположим, что векторы $\xi_1, \dots, \xi_M \in R^{n_1}$, представляющие образы, которые необходимо запомнить, не ортогональны, а только линейно независимы. Мы хотим сделать их ортогональными не меняя самих образов. Покажем, как этого можно добиться добавляя $L \geq M$ новых размерностей. Новые компоненты расширенных векторов будем обозначать v_k . Удобно рассматривать их как элементы пространства R^L . Задача состоит в том, чтобы создать ортогональные векторы $\zeta_k = \{\xi_k, v_k\} \in R^n$, $n = n_1 + L$. Компоненты v_k должны удовлетворять соотношению

$$(\xi_k \cdot \xi_j)_{n_1} + (v_k \cdot v_j)_L = A_k \delta_{kj}, \quad k, j = 1, \dots, M \quad (15)$$

Здесь $(\cdot)_{n_1}$ и $(\cdot)_L$ — скалярные произведения соответственно в подпространствах ξ и v . Покажем, что для достаточно больших A_k решение задачи существует.

Удобно рассматривать векторы v_k как столбцы матрицы V размером $L \times M$. Пусть матрица имеет вид

$$V = \begin{pmatrix} d & v_{12} & v_{13} & v_{14} & \dots & v_{1M} \\ & d & v_{23} & v_{24} & \dots & v_{2M} \\ & & d & v_{34} & \dots & \vdots \\ & & & d & \dots & \vdots \\ & & & & \ddots & v_{M-1,M} \\ & & 0 & & & d \end{pmatrix}.$$

Тогда условия (15) при $k = 1$ будут иметь вид

$$(\xi_1 \cdot \xi_j) + d \cdot v_{1j} = A_1 \delta_{1j},$$

откуда

$$v_{1j} = -(\xi_1 \cdot \xi_j) / d, \quad j = 2, \dots, M,$$

$A_1 = |\xi_1|^2 + d^2$. Таким образом, мы построили первый столбец и первую строку V и нашли вектор $\zeta_1 = \{\xi_1, v_1\}$, ортогональный ко всем другим ζ_k , независимо от выбора других компонент v_k . Затем делаем остальные векторы ортогональными к ζ_1 :

$$(\xi_2 \cdot \xi_j) + v_{12}v_{1j} + d \cdot v_{2j} = A_2 \delta_{2j},$$

и

$$v_{2j} = -\frac{(\xi_2 \cdot \xi_j) + v_{12}v_{1j}}{d}, \quad j = 2, \dots, M,$$

$A_2 = |\xi_2|^2 + v_{12}^2 + d^2$. После повторения этой процедуры $M - 1$ раз мы получим все компоненты v_k . Это доказывает ортогональность расширенных векторов ζ_k .

Тестовые расчёты подтвердили работоспособность этой схемы распознавания, а также то, что она позволяет запомнить вплоть до N линейно независимых образов и правильно распознать их. Полученный метод распознавания можно рассматривать как способ реализации метода подпространственной классификации образов [59].

8 Заключение. Выводы и перспективы

Мы описали некоторые динамические нейронные сети, производящие обработку информации. У них есть входной сигнал X и выходной $Y = \varphi^t(X)$, где φ — решение дифференциальных или разностных уравнений динамики сети. За исключением задачи моделирования временных рядов [45], обычно интересуются инвариантными множествами φ , аттракторами. Один и тот же результат можно получать разными способами, но обычно ищут наиболее эффективный, хотя эффективность часто относительна и зависит от доступного инструментария. Тем не менее, схема начальные данные \rightarrow аттрактор (или параметры \rightarrow аттрактор) не универсальна, так как данные могут поступать постепенно. Английская поговорка гласит, “if it quacks like a duck (временная эволюция), walks like a duck (временная эволюция), and has web feet (инвариантное свойство), then it is the duck”. Иногда подобные задачи можно решать, преобразуя их в обычные, записывая временные процессы, которые затем можно представить как векторы большой размерности. Однако в настоящее время их обычно решают методами искусственного интеллекта, а не с помощью нейронных сетей. Можно задать вопрос, как можно было бы расширить область применимости нейронных сетей и сделать динамический хаос полезным для их работы?

Мы рассмотрели ряд успешных применений и интересных подходов в области возможных будущих применений динамического хаоса в нейронных сетях в разделе 4. Однако они не расширяют области применимости существующих нейронных сетей. Если от сети требуется вполне определённый ответ (не зависящая от времени величина Y), какова может быть роль хаоса, дающего непредсказуемые результаты? С точки зрения определённого конечного результата, изолированная хаотичкая сеть не слишком полезна, за исключением случаев, когда надо избавиться от сходимости к ложным аттракторам.

Однако если нейронная сеть — это часть глобальной системы, польза может оказаться существенной. Назовём нейронную сеть “встроенной сетью”, если она образует управляющую часть сложной автономной системы. Встроенная нейронная сеть должна решать новый тип задач, нехарактерный для изолированных сетей: она должна учиться, взаимодействуя с окружающим “миром”. Примером может служить обучение поощрением, рассмотренное в разделе 6. В таких задачах иногда неважно, является ли выходной сигнал сети предсказуемым или нет. Зачастую автономный агент не имеет хорошо поставленной задачи; у него есть только некоторые критерии успеха. Как мы показали в этой статье, в интерактивном обучении у хаоса есть минимум одна ниша. Может быть, есть и другие.

Можно сделать вывод, что встроенные нейронные сети — это новый интересный объект исследований с точки зрения нелинейной динамики. Единственная проблема заключается в том, что для изучения встроенной сети необходима управляемая система. В современной хаотической динамике многие ключевые концепции были сформулированы при помощи простейших хаотических систем, таких как логистическое отображение, системы Хенона и Лоренца. Мы надеемся, что в будущем появятся и ”стандартные” системы и задачи управления для исследования встроенных сетей. Вероятнее всего, такие задачи будут связаны с робототехникой и направлением, получившем название “Artificial Life” — искусственная жизнь (см., например, обзорную книгу [73]). Небольшие автономные роботы или ”твари”, управляемые простыми нейронными сетями, строятся уже в течение ряда лет. Как правило, создававшиеся искусственные автономные создания были лишены способности обучаться. Необходимая конфигурация нейронной сети возникает в результате использования генетических или эволюционных алгоритмов. Поведение таких созданий иногда выглядит хаотическим, однако, насколько нам известно, детально оно не исследовалось. Теперь возникает задача построения искусственных созданий, которые могли бы как эволюционировать, так и обучаться. И вполне вероятно, что динамический хаос будет типичным состоянием для управляющих ими нейронных сетей [13], или же такие сети будут располагать хаотическими подсистемами. По-видимому, встроенные нейронные сети и искусственная жизнь — это естественная ниша для нелинейной динамики и хаоса.

Динамические системы описывают огромную область природных явлений, в то время как нейронные сети покрывают лишь их небольшую часть. Есть микроскопические, мезоскопические, макроскопические системы. Концепция обработки информации в нейронных сетях несомненно будет расширять область своей применимости в сравнении с сегодняшним состоянием. Ближайшей перспективой представляется её расширение на область квантовых вычислений, основанных на гамильтоновых нейросетях. В настоящей работе мы привели простейший пример классической гамильтоновой нейронной сети. Сейчас о подобных сетях известно очень немного. Как нам кажется, следует ожидать расширения работ в этом направлении.

Работа выполнена в Летбриджском университете (University of Lethbridge, Canada). Она поддержана грантом М.К.А. от Defence Research Establishment Suffield, контракт No. W7702-8-R745/001/EDM. Использовались вычислительные мощности, предоставленные Multimedia Advanced Computational Infrastructure (MACI).

Nonlinear dynamics of information processing in neural networks

A.B. Potapov and M.K. Ali

We consider a number of possible roles of complex dynamics and chaos in information processing by neural networks. First, we review the principles of operation for some well-known neural networks, and then discuss a approaches to using chaos in neural networks. Our main goal is to present a novel view of the problem of chaos in information processing. We demonstrate that chaos emerges naturally when a neural network forms a controlling part of a more complex system. We show that such neural networks can enhance efficiency by using chaos for explorations in a method known as Reinforcement Learning. A discussion on Hamiltonian neural networks is also presented.

Литература

- [1] Adachi M., Aihara K. Associative dynamics in a chaotic neural network. *Neural Networks*, **10** (1997) 83–98.
- [2] Aihara K., Takabe T., Toyoda M., “Chaotic neural networks”, *Phys. Lett. A* **144**, 333–340 (1990).
- [3] Albers D.J., Sprott J.C., Dechert W.D., “Routes to chaos in neural networks with random weight”, *Int. J. Bifurc. Chaos* **8**, 1463–1478 (1998).
- [4] Andreyev Yu.V., Dmitriev A.S., Chua L.O., Wu C.W., “Associative and random access memory using one-dimensional maps”, *Int. J. Bifurc. Chaos* **2**, 483–504 (1992).
- [5] Andreyev Yu.V., Dmitriev A.S., Starkov S.O., “Information processing in 1-D systems with chaos”, *IEEE Trans. on Circuits and Systems* **44**, 21–28 (1997).
- [6] Aradi I., Barna G., Érdi P., Gröbler T. Chaos and learning in the olfactory bulb. *Int. J. Intelligent Systems* **10** (1995) 89–117.
- [7] Arbib M.A. *Brains, machines, and mathematics*. New York : Springer-Verlag, 1987, xvi+202 p.
- [8] Babloyantz A., Destexhe A., Low-dimensional chaos in an instance of epilepsy. *Proc. Natl. Acad. Sci. USA*, **83**, 3513 (1986).
- [9] Baird B., Nonlinear dynamics of pattern formation and pattern recognition in the rabbit olfactory bulb. *Physica D*, **22**, 150 (1986).
- [10] Baird B., Eeckman F. A normal form projection algorithm for associative memory. in: [43], p.135–166.
- [11] Banerjee S., Yorke J.A., Grebogi C., “Robust chaos”, *Phys. Rev. Lett* **80**, 3049–3052 (1998).
- [12] Barto A.G., Sutton R.S., Anderson C.W., “Neuronlike adaptive elements that can solve difficult control problems” *IEEE Trans. on Systems, Man, and Cybernetics* **SMC-13**, 834–846 (1983).
- [13] Barton S.A., “Two-dimensional movement controlled by a chaotic neural network”, *Automatica* **31**, 1149–1155 (1995).
- [14] Bellman R., *Dynamic Programming* (Princeton University Press, Princeton, 1957).
- [15] Bertsekas D.P., Tsitsiklis J.N., *Neuro-Dynamic Programming* (Athena Scientific, Belmont, 1995).
- [16] Bishop C.M. *Neural networks for pattern recognition*. Oxford: Clarendon Press, 1995, xvii+482pp.

- [17] Carpenter G.A. Neural network models for pattern recognition and associative memory. *Neural Networks* **2** (1989) 243–257; опубликовано также в [18], 2–33.
- [18] Carpenter G.A., Grossberg S. (eds.) *Pattern recognition by self-organizing neural networks*. MIT Press: Cambridge, MA, 1991, 691p.
- [19] Carpenter G.A., Grossberg S. A massively parallel architecture for a self-organizing neural pattern recognition machine. *Computer Visions, Graphics and Image Processing*, **37** (1987) 54–115; also [18], 316–382.
- [20] Carpenter G.A., Grossberg S. ART2: self-organization of stable category recognition codes for analog input patterns. *Applied Optics*, **26** (1987) 4919–4930; опубликовано также в [18], 398–423.
- [21] Carpenter G.A., Grossberg S. ART3: hierarchical search using chemical transmitters in self-organizing pattern recognition architectures. *Neural Networks* **3** (1990) 129–152; опубликовано также в [18], 453–499.
- [22] Carpenter G.A., Grossberg S., Reynolds J. ARTMAP: supervised real-time learning and classification of nonstationary data by a self-organizing neural network. *Neural Networks* **4** (1991) 565–588; опубликовано также в [18], 503–544.
- [23] Chang-song Z., Tian-lin C., Wu-qun H. Chaotic neural network with nonlinear self-feedback and its application in optimization. *Neurocomputing*, **14** (1997) 209–222.
- [24] Chen L., Aihara K., “Chaotic simulated annealing by a neural network model with transient chaos”, *Neural Networks*, Vol. 8, pp. 915-930, 1995
- [25] Cherkassky V., Mulier F. *Learning from data. Concepts, theory and methods*. NY etc: Wiley, 1998.
- [26] Dmitriev A.S., Kuminov D.A. Chaotic scanning and recognition of images in neuron-like systems with learning. *J. of Communications Technology and Electronics*, **39** (1994) 118–127.
- [27] Domany E., van Hemmel J.L., Schulten K. (eds.) *Models of neural networks I*. Berlin: Springer, 1995.
- [28] Eckmann J.-P., Ruelle D., “Ergodic theory of chaos and strange attractors” *Rev. Mod. Phys.* **57**, 617–656 (1985).
- [29] Farmer D.J. A rosetta stone for connectionism. *Physica D*, **42** (1990) 153–187.
- [30] Freeman W.J. Qualitative overview of population neurodynamics. in: *Neural Modeling and Neural Networks* (Pergamon Studies in Neuroscience, No 11) F. Ventriglia, ed., Pergamon, 1993, 185–215.
- [31] Freeman W.J. Role of chaotic dynamics in neural plasticity. in: J. van Pelt, M.A. Corner, H.B.M. Uylings, F.H. Lopes da Silva (eds.) *Progress in Brain Research*, **102**, Springer, 1994.

- [32] Freeman W.J. Simulation of Chaotic EEG patterns with a dynamic model of the olfactory system. *Biol. Cybern.*, **56** (1987) 139–150.
- [33] Freeman W.J., “The physiology of perception” *Scientific American* **264/2**, 78–85 (1991).
- [34] Freeman W.J. Tutorial on neurobiology: from single neurons to brain chaos. *Int. J. Bifurc. Chaos*, **2** (1992) 451–482.
- [35] Freeman W. J., Yao Y., Burke B., Central pattern generating and recognizing in olfactory bulb: a correlation learning rule. *Neural Networks*, **1**, 277 (1988).
- [36] Gadaleta S., Dangelmayr G., “Optimal chaos control through reinforcement learning.” *Chaos* **9**, 775–788 (1999).
- [37] Gardner E. The space of interactions in neural network models. *J.Phys. A*, **21** (1988) 257–270.
- [38] Glanz J., Sharpening the senses with neural “noise”. *Science* **277**, 1758 (1997).
- [39] Grossberg S. Nonlinear neural networks: principles, mechanisms, and architectures. *Neural Networks* **1** (1988) 17–61; also [18], 36–109.
- [40] Guckenheimer J., Holmes P., *Nonlinear Oscillations, Dynamical Systems and Bifurcations of Vector Fields* (Springer, NY, 1983).
- [41] Guevara M.R., Glass L., Mackey M.C., Shrier A., “Chaos in neurobiology” *IEEE Trans. on Systems, Man and Cybernetics* **SMC-13**, 790–798 (1983).
- [42] Haken H. *Information and self-organization*. Berlin:Springer-verlag, 1988.
- [43] Hassoun M.H., ed. *Associative Neural Memories*, NY, Oxford: OUP, 1993, 350p.
- [44] Hayakawa Y., Marumoto A., Sawada Y., “Effects of the chaotic noise on the performance of a neural network model for optimization problems”, *Phys. Rev. E*, **51(4)**, R2693–R2696 (1995).
- [45] Haykin S. *Neural networks: a comprehensive foundation*. Macmillan: N.Y. etc., 1994.
- [46] Hecht-Nielsen R. *Neurocomputing*. Reading, MA etc.: Addison-Wesley, 1990, xiii+433pp.
- [47] Hecht-Nielsen R. Counterpropagation networks. *Applied optics*, **26** (1987) 4979–4984; опубликовано также в [18], 264–276.
- [48] Ho C.Y., Kurokawa H., A learning algorithm for oscillatory cellular neural networks. *Neural Networks* **12** (1999) 825–836.
- [49] Hopfield J.J. Neural networks and physical systems with emergent collective computational abilities. *Proc. Natl. Acad. Sci. USA*, **79** (1982) 2554–2558.
- [50] Hopfield J.J. Neurons with graded response have collective computational properties like those of two-state neurons. *Proc. Natl. Acad. Sci. USA*, **81** (1984) 3088–3092.

- [51] Hopfield J.J., Tank D.W. “Neural” computation of decisions in optimization problems. *Biol. Cybern.*, **52** (1985) 141–152.
- [52] Hoppensteadt F.C., Izhikevich E.M., Oscillatory neurocomputers with dynamic connectivity. *Phys. Rev. Lett.* **82** (1999) 2983–2986.
- [53] Hoppensteadt F. C., Izhkevich E. M., *Weakly Connected Neural Networks* (Springer-Verlage, New York, 1997).
- [54] Ishii S., Fukumizu K., Watanabe S. A network of chaotic elements for information processing. *Neural Networks*, **9** (1996) 25–40.
- [55] Izhikevich E.M., Malinetskii G.G. A possible role of chaos in neurosystems. *Sov. Phys.-Dokl. (USA)*, **37** (1992) 492–495.
- [56] Jibu M., Yassue K., in: *Rethinking Neural Networks*, ed. K. H. Pribram, Lawrence Erlbaum Associates, Inc., Publishers , Hillsdale New Jersey, USA (1993)
- [57] Kaelbling L.P., Littman M.L., Moore A.W., “Reinforcement learning: A survey” *J. Artificial Intelligence Research* **4**, 237–285 (1996).
- [58] Kosko B. *Neural networks and fuzzy systems*. Englewood Cliffs: Prentice Hall, 1992, xxvii+449pp.
- [59] Kohonen T. *Self-organizing maps*. Berlin:Springer, 1997, 448p.
- [60] Kohonen T. The “neural” phonetic typewriter. *Computer* **21** (1988) 11–22; опубликовано также в [18], 239–259.
- [61] Kürten K.E., Clark J.W., “Chaos in neural systems”, *Phys. Lett. A* **114**, 413–418 (1986).
- [62] Kwok T., Smith K.A., A unified framework for chaotic neural -network approach to combinatorial optimization”, *IEEE Trans. Neural Networks* **10** (1999) 978–981 (and references therein).
- [63] McCulloch W.S., Pitts W., A logical calculus of the ideas immanent in nervous activity. *Bull. Math. Biophys.* **5** (1943) 115–133.
- [64] Michie D., Chambers R.A., “Boxes: an experiment in adaptive control” in: E. Dale, D. Michie, eds. *Machine Learning 2* (American Elsevier, N.Y., 1968), p.137–152.
- [65] Miller W.T., Sutton R.S., Werbos P.J., eds. *Neural Networks for Control* (MIT Press, Cambridge, London, 1990).
- [66] Morita M. Associative memory with nonmonotone dynamics. *Neural Networks*, **6** (1993) 115–126.
- [67] Nakagawa M., A chaos associative model with a sinusoidal activation function. *Chaos, Solitons and Fractals* **10** (1999) 1437–1452.

- [68] Nielsen M.A., Chuang I.L., *Quantum Computing and Quantum Information*. (Cambridge University Press 2000).
- [69] Nishii J., A learning model for oscillatory networks. *Neural Networks* **11** (1998) 249–257.
- [70] Pasemann F., “A simple chaotic neuron”, *Physica D* **104**, 205-211 (1997).
- [71] Penrose R. *Shadows of the mind*. Vintage, 1995.
- [72] Personnaz L., Guyon I., Dreyfus G., *J. de Physique Lettres* **46** (1985) L359–L365.
- [73] Pfeifer R., Scheier C., *Understanding Intelligence* (The MIT Press, Cambridge MA, 1999).
- [74] Pineda F.J. Generalization of backpropagation to recurrent neural networks. *Phys. Rev. Lett.*, **59** (1987) 2229–2232.
- [75] Potapov A., Ali M.K., “Robust chaos in neural networks”, *Phys. Lett. A* **277** (2000) 310–322.
- [76] Potapov A.B., Ali M.K., Learning, Exploration and Chaotic Policies. *Int. J. Modern Physics C* **11** (2000) 1455–1464.
- [77] Potapov A.B., Ali M.K., Chaotic neural control. *Phys. Rev. E* **63** (2001), April.
- [78] Pribram K.H.. *Brain and perception*. Hillsdale: Lawrence Erlbaum Associates, 1991, xxix+388p.
- [79] Rumelhart D.E., McClelland J.L., eds., *Parallel Distributed Processing: Explorations in the Microstructure of Cognition* (vol. 1, 2). The MIT Press, 1986.
- [80] Ryan T.W., Winter C.L. Variations on adaptive resonance. *IEEE First International Conference on Neural Networks*, M. Caudill and C. Butler, eds., San Diego: IEEE, 1987, 767-775; опубликовано также в [18], 385–396.
- [81] Sarle, W.S., ed. *Neural Network FAQ*.
URL: <ftp://ftp.sas.com/pub/neural/FAQ.html>.
- [82] Sharkey A.J.C., ed. Combining artificial neural nets. Ensemble and modular multi-net systems. NY:Springer, 1999, 304p.
- [83] Sutton R.S., Barto A.G. *Reinforcement learning*. Cambridge and London: A Bradford Book and MIT Press, 1998, xviii+322pp. Многие материалы книги доступны в Интернете, <http://envy.cs.umass.edu/~rich/book/the-book.html>.
- [84] Sutton R.S., Barto A.G., “Toward a modern theory of adaptive networks: expectation and prediction” *Psychological review* **88**, 135–170 (1981).
- [85] Skarda C. A., Freeman W. J., How brains make chaos in order to make sense of the world. *Behav. Brain Sci.*, **10**, 161 (1987).

- [86] Tan Z., Ali M.K., Pattern recognition in a neural network with chaos. *Phys. Rev. E*, **58**, 3649 (1998).
- [87] Tan Z., Ali M.K., Associative memory using synchronization in a chaotic neural network. *Int. J. Modern Physics C* **12** (2001).
- [88] Tan Z., Ali M.K., Pattern recognition with stochastic resonance in a generic neural network. *Int. J. Modern Physics C* **12** (2001).
- [89] Tan Z., Hepburn B.S., Tucker C., Ali M.K. Pattern recognition using chaotic neural networks. *Discrete Dynamics in Nature and Society*, **2** (1998) 243–247.
- [90] Tesauro G., Practical issues in temporal difference learning. *Machine Learning* **8** (1992) 257–277
- [91] Thrun S.B., “The role of exploration in learning control”, in: D.A. White, D.A. Sofge, eds., *Handbook of Intelligent Control. Neural, Fuzzy, and Adaptive Approaches* (Van Nostrand Reinhold, NY, 1992).
- [92] Touzet C., “Neural reinforcement learning for behaviour synthesis.” *Robotics and Autonomous Systems* **22**, 251–281 (1987).
- [93] Tsuda I., Dynamic link of memory — chaotic memory map in nonequilibrium neural networks. *Neural Networks*, **5**, 313 (1992).
- [94] Ventura D., Martinez T., Quantum Associative Memory. xxx.lanl.gov, quant-ph/9807053 (1998).
- [95] Wang X.-J., Rinzler J., *Handbook of Brain and Neural Networks*, ed. M. Arbib, (MIT Press, Cambridge, 1995).
- [96] Weigend A.S., Gershenfeld N.A., eds., *Time series prediction: forecasting the future and understanding the past*. Addison-Wesley, 1994.
- [97] White D.A., Sofge D.A., eds. *Handbook of Intelligent Control. Neural, Fuzzy and Adaptive Approaches* (Van Nostrand Reinhold, N.Y., 1992).
- [98] Williams C.P., Clearwater S.H., *Exploration in Quantum Computing*. (Springer-Verlag, New York 1998).
- [99] Yao Y., Freeman W.J. Model of biological pattern recognition with spatially chaotic dynamics. *Neural Networks*, **3** (1990) 153–170.
- [100] Yao Y., Freeman W.J., Burke B., Yang Q., Pattern recognition by a distributed neural network: an industrial application. *Neural Networks*, **4**, 103 (1991).