

Санкт-Петербургский государственный университет
Факультет прикладной математики - процессов управления
Кафедра компьютерных технологий и систем

Климов Николай Сергеевич

Магистерская диссертация

**ИСПОЛЬЗОВАНИЕ МОДЕЛИ ИЕРАРХИЧЕСКОЙ
ТЕМПОРАЛЬНОЙ ПАМЯТИ В ЗАДАЧАХ
МАШИННОГО ОБУЧЕНИЯ**

Направление 02.04.02

Фундаментальная информатика и информационные технологии
Магистерская программа автоматизация научных исследований

Научный руководитель
кандидат физ.-мат.наук
доцент
Коровкин М.В.

Санкт-Петербург
2016

Оглавление

Введение	3
Актуальность задачи	4
Постановка задачи	5
1 Обзор существующих решений	7
1.1 Классификаторы с учителем	7
1.1.1 Нейронные сети прямого прохождения	7
1.1.2 Байесовские сети	8
1.2 Методы ближайших соседей	9
1.3 Кластеризация	10
1.4 Рекуррентные нейронные сети	10
1.5 Общий обзор моделей и преимущества НТМ	11
2 Биологические предпосылки модели	12
2.1 Биологические структуры и их модели	15
2.1.1 Нейрон	15
2.1.2 Колонка	16
3 Описание модели	18
3.1 Обзор НТМ	18
3.2 Пространственные разреженные представления	21
3.3 Энкодеры	24
3.3.1 Числовой энкодер	25
3.3.2 Энкодер категорий	25
3.3.3 Временной энкодер	26
3.3.4 Геопространственный энкодер	26
3.4 Клетка НТМ	28
3.5 Пространственный группировщик	31
3.6 Темпоральный группировщик	32
3.7 Поиск аномалий	34
4 Проведение эксперимента	35
4.1 Входные данные	35
4.2 Конфигурация НТМ	35

4.3	Оборудование и программное обеспечение	36
4.4	Результаты эксперимента	36
	Заключение	40
	Литература	43
	Используемые аббревиатуры	43

Введение

В последнее десятилетие мы видим все большее и большее количество приложений искусственного интеллекта, их можно встретить везде: от ботов в компьютерных играх до приложений распознавания лиц на фотографиях. Одним из недостатков таких алгоритмов является их неуниверсальность. Метод, применяемый для решения одной задачи, зачастую не подходит для решения другой. Для того, чтобы мы могли конструировать более лучшие системы, нужен новый подход, из которого мы сможем получить более универсальный алгоритм. Одна такая универсальная система давно нам известна - это мозг млекопитающих, вернее неокортекс.

Несмотря на то, что модели отдельных участков нейронной сети появились уже более полувека назад, универсальной теории работы неокортекса не было довольно долго. Наверное одной из первых более менее удачной попыток построения такой модели была книга *On intelligence*, опубликованная Джеффом Хокинсом и Сандрой Балакесли в 2004 году [1]. В этой книге Хокинс приводит последние (на тот момент) исследования работы неокортекса и закладывает базу для модели иерархической темпоральной памяти (*Hierarchical Temporal Memory*, далее в тексте будет применяться аббревиатура НТМ).

Хокинс утверждает, что его методы станут основой для "по-настоящему интеллектуальных машин". Он использует довольно определенное понятие интеллекта, основанное на предсказании. Согласно ему, система может называться интеллектуальной, если она непрерывно делает предсказания окружающего мира, сравнивает эти предсказания с показаниями сенсоров и изменяет свои параметры для получения новых, более точных предсказаний. Подобная система постоянно улучшает внутреннюю модель мира с целью лучшего понимания того, что в нем происходит.

Цель данной работы заключается в разборе предложенной модели и

анализе её применимости в задачах машинного обучения. Так как машинное обучение – очень обширная область, было решено сконцентрироваться на определенной задаче - поиск аномалий в GPS данных. Именно на подобных задачах, как будет показано далее, НТМ работает лучше всего.

Актуальность задачи

В наши дни почти у каждого человека в кармане есть аппарат, позволяющий с довольно высокой точностью определить свое местоположение. Речь идет о смартфонах и GPS приемнике, установленных в них. С распространением облачных технологий не представляет труда и автоматизация процесса сбора этих данных и сохранения в облаках для последующей обработки.

Поиск аномалий в передвижении объектов может быть использовано в разных областях человеческой деятельности. Несколько примеров таких областей:

- Безопасность авиатранспортировки. Наблюдение за отклонениями параметров полета от нормы.
- Безопасность морских перевозок. Обнаружение аномальных маршрутов перевозок, заходов в порт, поломок судна, пиратства.
- Уход за животными и членами семьи. Поиск аномалий в передвижении детей или людей, которые нуждаются в присмотре, например больных альцгеймером.
- Мониторинг сотрудников фирмы.
- Мониторинг оборудования.

Во всех этих областях нахождение аномалии может помочь оперативно устранить какую-то проблему и даже потенциально сохранить человеческую жизнь.

Постановка задачи

В данной работе будет исследоваться применимость НТМ к задаче поиска аномалий в GPS данных. GPS данные представляют собой полученные с помощью GPS спутников координаты на поверхности Земли, время получения этих координат, а также различная вспомогательная информация вроде скорости, направления движения, количества спутников и т.п. В нашей задаче нам будут интересны только значения широты, долготы, скорости и времени.

Вообще говоря, довольно сложно формально определить понятие аномалии или аномальных данных. Чаще всего разделяют три вида аномалий [2]:

- 1 Если точка, по отношению к другим данным, является аномальной, то её называют **точечной аномалией** (point anomaly).
- 2 Если точка сама по себе не является аномальной, но в контексте других данных ведет себя аномально, то её называют **контекстной аномалией** (contextual anomaly). Рис. 1.
- 3 Если группа точек является аномалией только по отношению ко всем данным, но её называют **коллективной аномалией** (collective anomalies). Рис. 2.

В данной работе будем считать аномальностью существенное отклонение пути от какого-то привычного маршрута или сильное изменение скорости передвижения. Это можно считать как контекстной, так и коллективной аномалией (зависит от того хотим ли мы найти аномалии отдельных точек маршрута или целого пути). В данной работе будут искаться аномалии отдельных точек, в зависимости от уже пройденного маршрута, следовательно будет рассматриваться контекстный тип аномалий.

В ходе работы сеть НТМ будет обучаться таким образом, чтобы при поступлении нового сигнала (новой точки GPS пути) она вычисляла коэффициент аномальности (anomaly score) в пределах от 0 до 1. Соответственно точки, у которых этот коэффициент выше определенного порога будем считать аномалией.

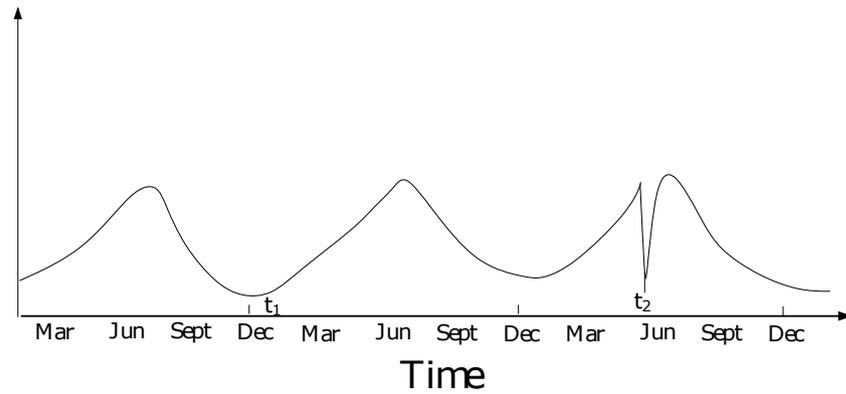


Рис. 1: Пример контекстной аномалии. Значение функции в точках t_1 и t_2 одинаковые, поэтому невозможно определить аномальны ли они, только по их значению. Однако по контексту можно понять, что точка t_2 ведет себя аномальным образом.

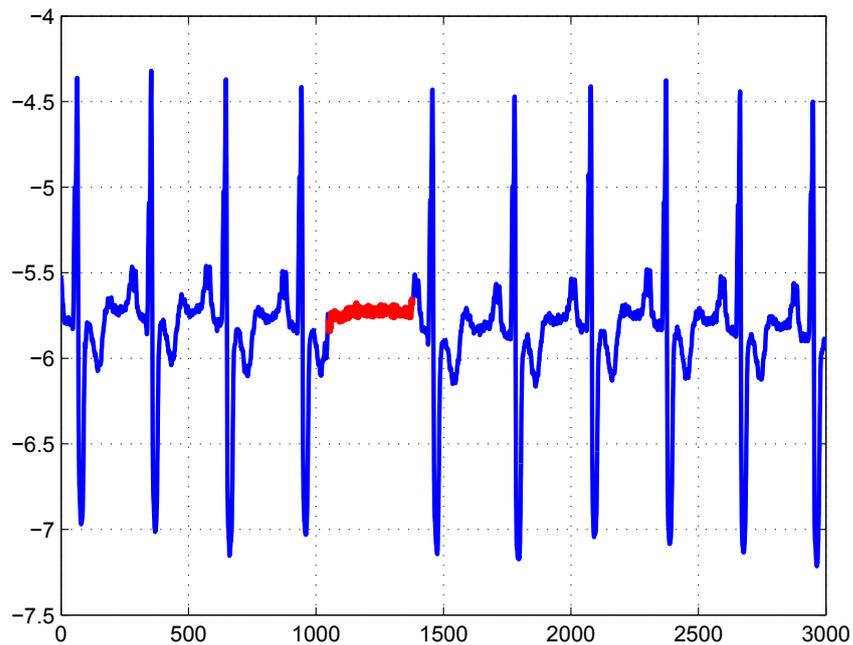


Рис. 2: Пример коллективной аномалии в электрокардиограмме. В данном случае, взятые несколько точек из выделенного участка не являются аномальными сами по себе, аномальностью является то, что сердце находится в этом положении слишком долго. Чтобы выявить это нужно просматривать большое окно данных (не малую окрестность вокруг изучаемых точек).

Глава 1

Обзор существующих решений

В данном разделе будет представлен краткий обзор различных техник применяемых в задачах детектирования аномалий, после этого будет описана возможность применения этих методов в поставленной в этой работе задаче.

1.1 Классификаторы с учителем

В общем случае, если входные данные имеют метки классов (нормальные данные и аномалии), то простейший способ найти аномалии – использовать методы машинного обучения с учителем.

1.1.1 Нейронные сети прямого прохождение

Изначально нейронные сети прямого прохождение были воодушевлены принципами работы мозга. Однако в этой модели используют очень упрощенную модель нейрона и синапса. Существует огромное количество видов нейронных сетей, однако от начальной идеи моделирования работы мозга в итоге отказались, её заменили различные статистические методы.

Основные принципы работы нейронной сети следующие:

- Нейроны организованны в слои
- Соединения нейронов имеют коэффициенты связей

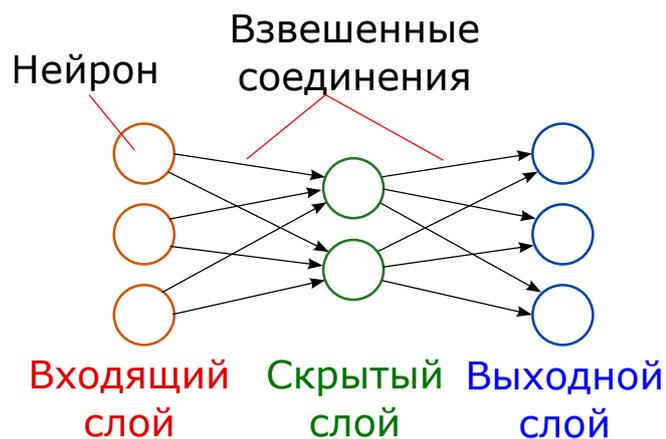


Рис. 1.1: Нейронная сеть прямого распространения

- Выход нейрона является функцией от взвешенной суммы входа нейрона
- Веса и параметры сети должны быть настроены до её использования

Для поиска аномалий нейронными сетями необходимо обучить сеть на нормальных данных. После обучения на вход сети подаются данные. Если сеть приняла эти данные, то они считаются нормальными, иначе аномальными [3].

Нейронные сети – очень мощный и гибкий инструмент, используя методы скользящего окна можно даже получить предсказания последовательностей сигналов [4]. Однако для обучения таких сетей требуется огромное количество данных, а использование метода обучения без учителя сильно ограничено.

1.1.2 Байесовские сети

Байесовские сети (или сети доверия) используют причинно-следственные связи и вероятностные распределения, составляя направленный ациклический граф из случайных переменных (рис 1.2)

Подход к определению аномалий такой же, как и у нейронных сетей [5]. Байесовские сети обучаются на нормальных классах, далее если итоговая вероятность соотнесения к классу мала, данные считаются аномальными. Явным недостатком такого подхода является то, что человек

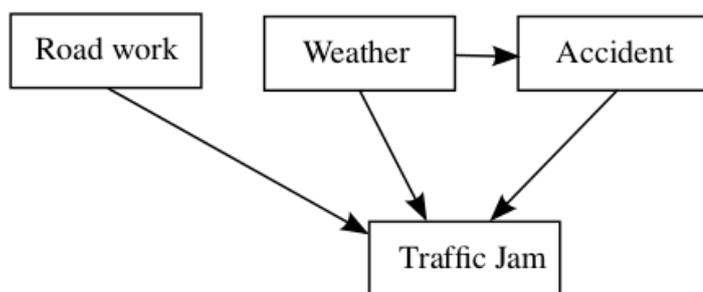


Рис. 1.2: Пример баесовской сети

должен сам составить топологию сети, исходя из знания предметной области, что зачастую само по себе является крайне сложной задачей.

1.2 Методы ближайших соседей

Методы данного типа предполагают, что нормальные данные очень плотно лежат в пространстве сигналов, а аномалии расположены далеко от соседних сигналов. В этом подходе для каждого нового сигнала рассчитывается плотность соседей. Если плотность низкая, то сигнал признается аномалией [6]. Метрикой расстояния обычно используется обычная евклидова метрика, но могут быть и модификации.

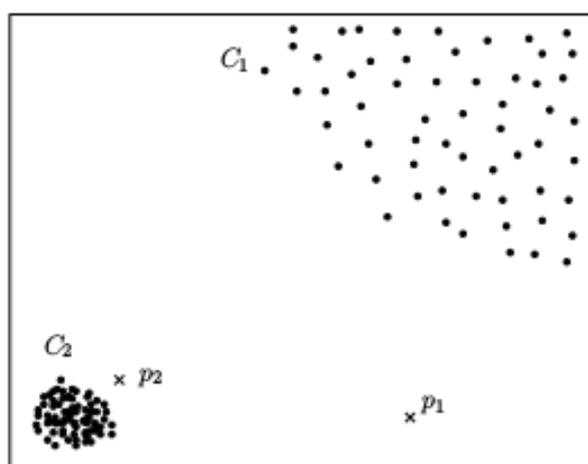


Рис. 1.3: Пример выявления аномалии методом ближайших соседей.

Существует огромное количество модификаций этого подхода. Преимуществами этого подхода можно назвать то, что в нем не делается предположений о задаче, все выясняется только из данных. По этой же причине этот алгоритм довольно универсален, то есть подходит для разных типов

данных.

Недостатками этого подхода является слишком сильное предположение о данных. Если у нормальных данных будет мало соседей, и наоборот у аномальных данных соседей будет много, метод может давать ошибочные результаты. Так же при больших объемах данных метод работает слишком долго.

1.3 Кластеризация

В этом подходе используется предположение, что нормальные данные находятся близко от центроида своего кластера, а аномалии лежат далеко. Коэффициентом аномальности (anomaly score) в данном случае можно считать расстояние сигнала до ближайшего центроида кластера. Существует множество алгоритмов кластеризации, такие как самоорганизующиеся карты Кохонена, K-means, алгоритм максимального правдоподобия (EM алгоритм) [7].

Плюсами такого подхода можно назвать то, что обучение осуществляется без учителя, что алгоритм подходит для разных типов данных, а также высокую скорость алгоритмов. Однако если аномалии лежат близко к кластерам, такой подход будет давать ошибки.

1.4 Рекуррентные нейронные сети

Рекуррентные нейронные сети - модификация нейронных сетей прямого прохождения. Пример такой сети на рис. 1.4. Благодаря обратным связям сеть может запоминать предыдущее состояние и выявлять временные паттерны [8]. Однако такие сети трудно обучать. Чтобы избежать проблемы затухающего градиента при обучении методом обратного распространения ошибки [9] используют другие модели нейрона, например Long Short Term Memory (LSTM).

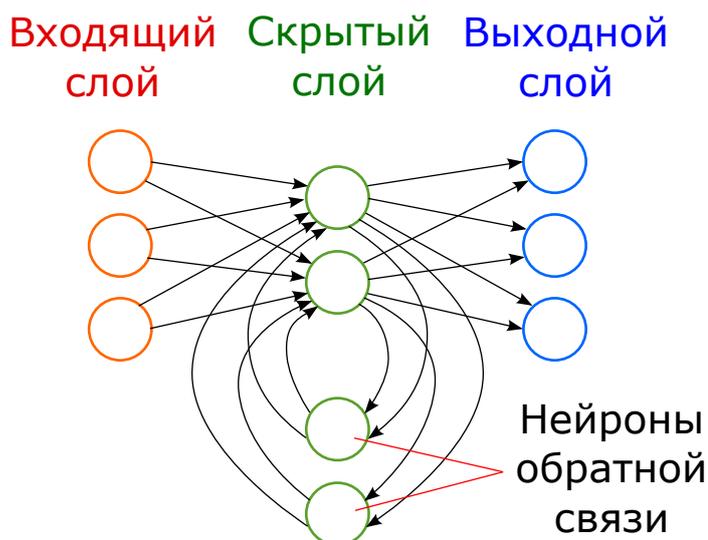


Рис. 1.4: Рекуррентная нейронная сеть

1.5 Общий обзор моделей и преимущества НТМ

Большинство приведенных методов, хоть и являются довольно популярными в поиске аномалий, однако могут находить только точечные аномалии. Без сложных модификаций контекстные аномалии большой последовательности данных могут находить только модели с памятью, то есть рекуррентные нейронные сети.

Основной недостаток этого метода заключается в том, что он требует большого количества данных для обучения и они больших вычислительных ресурсов. Достоинство НТМ по отношению к рекуррентным сетям, является в том, что НТМ лучше масштабируется, за счет того, что обратные связи представлены только внутри региона. Благодаря этому довольно легко понять как можно расширять сеть как вширь (добавляя колонок), так и в длину (добавляя слои). К тому же благодаря алгоритму SP, в НТМ нет "мертвых" колонок, которые не участвуют в работе сети.

Существенным недостатком обеих моделей – сложный выбор параметров сети, особенно в задачах обучения без учителя, где довольно сложно численно оценивать точность моделей.

Глава 2

Биологические предпосылки модели

Как уже было сказано выше, НТМ была создана в первую очередь как модель работы неокортекса, поэтому имеет смысл сделать краткий обзор биологических предпосылок этой модели.

Впервые идеи НТМ были озвучены в 2004 году в книге Джефа Хокинса *On Intelligence* [1]. В ней Хокинс писал о подходе разработки простой модели неокортекса не симулируя его составные части, а пытаясь понять какие процессы в биологической модели являются наиболее важными, чтобы моделировать только их. Нельзя сказать, что этот подход революционный, те же нейронные сети тоже упрощенная модель неокортекса.

Главная проблема в построении моделей мозга состоит в том, что у нейронов и нейронных сетей слишком много анатомических и функциональных деталей для воспроизводства, и понять что из этого действительно необходимо для модели нейронной сети (в вычислительном смысле) довольно сложно. Тут можно привести аналогию пришельцев, которые пытаются понять как устроены наши микропроцессоры. Главная суть микропроцессоров заключается в том, как из элементарных логических цепей составляются схемы сумматоров, ячеек памяти и т.п., которые далее, на более высоком уровне абстракции, переходят в схемы конвейеров или кеш-памяти. Причем такие логические цепи можно делать и другими средствами, без транзисторов. Однако все, что видят пришельцы - это транзисторы, которые переключаются каким-то хитрым образом, и понять как из транзисто-

ров получается конвейер операций задача нетривиальная.

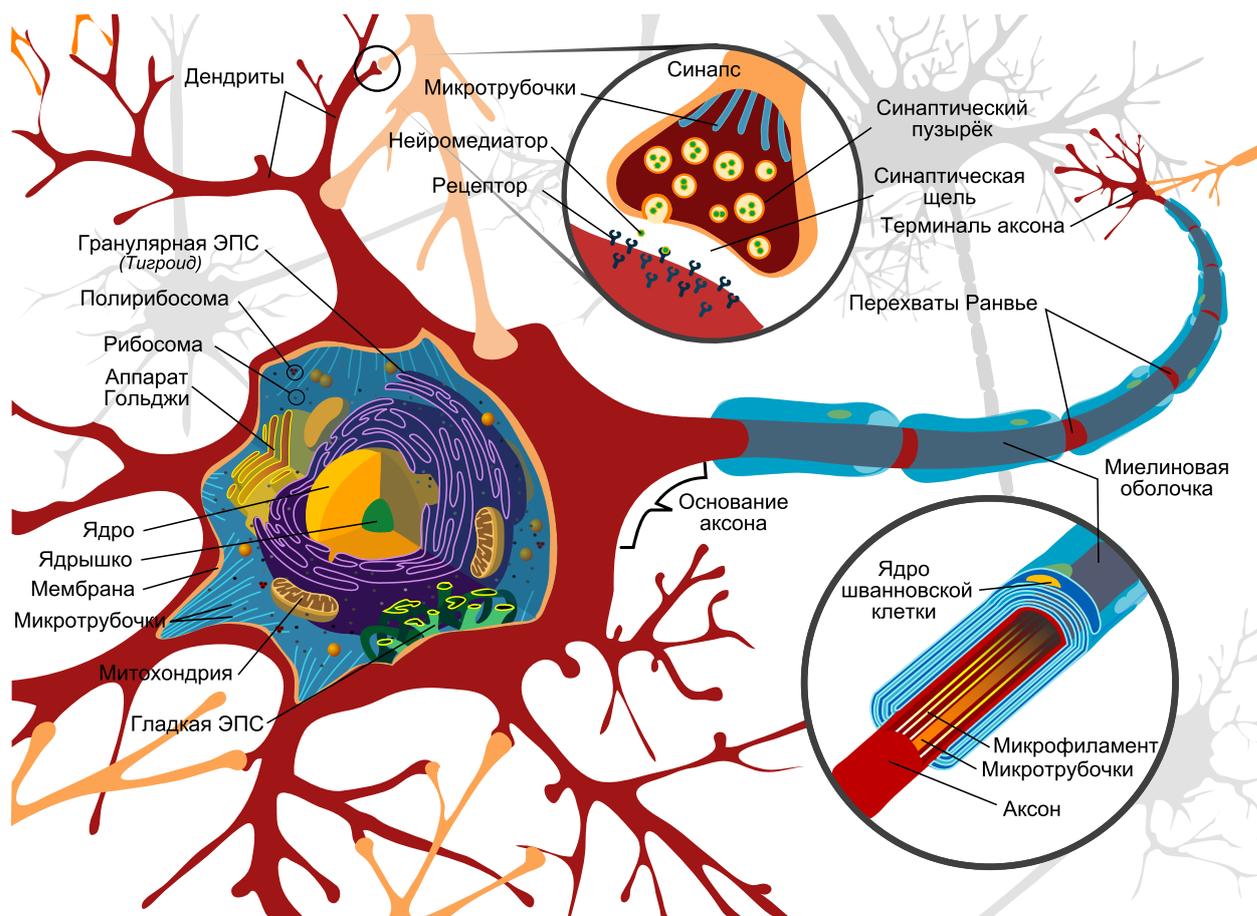


Рис. 2.1: Схема клетки нейрона. Все составные части клетки так или иначе влияют на её работу, а значит и на поведение нейронной сети в целом, вопрос в том насколько они влияют и можно ли ими пренебречь?

Классические нейронные сети представляют собой наиболее простейшую модель, без обратных связей и сложного поведения. По большому счету классические нейронные сети являются каскадом линейных классификаторов.

Джеф Хокинс в своей книге выделил несколько принципов, которым, по его мнению, обязана удовлетворять хорошая модель неокортекса:

1) Модель должна быть иерархической. Каждая сенсорная область коры головного мозга представляет собой иерархическую структуру. На нижние уровни иерархий поступают сигналы с сенсоров (с сетчатки глаза, с биполярных клеток уха и т.п.). Информация с нижних уровней обрабатывается и передается на верхние уровни. Рассмотрим этот поток сигналов на примере зрительного пути.

В первичную зону зрительная информация идет из таламуса (то есть

практически напрямую из зрительных нервов). В этой зоне происходит первичная обработка сигналов, заключающаяся в выделении границ на изображении [10]. Далее эта информация поступает в следующие области, где на более высоких уровнях иерархии обрабатывается более абстрактная информация. Например на рис. 2.2 показано как сигналы из первичной коры (попутно проходя через другие регионы) попадают в зоны отвечающие за определение местонахождения объектов (путь к теменной зоне) и в зону отвечающую на распознавание объектов (путь к височной доле). Более подробно об аргументации этого принципа можно прочитать в [11].

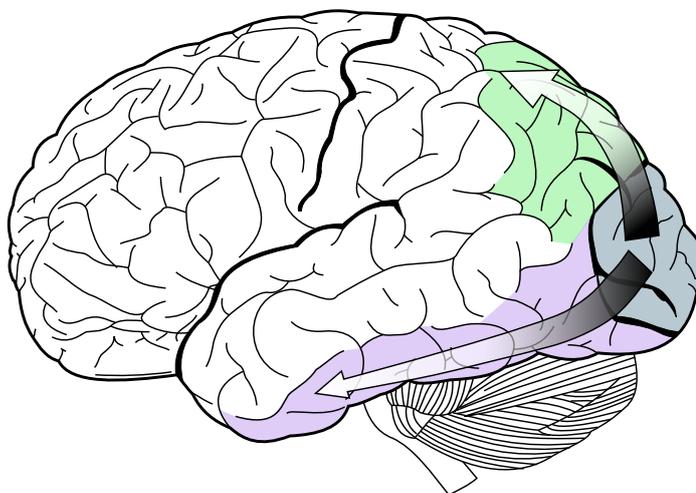


Рис. 2.2: 2 пути зрительного сигнала

2) В неокортексе не заложено "специальных" зон для зрительной или, например, для слуховой информации, поэтому таких зон не должно быть и в предполагаемой модели. Имеется в виду, что все части неокортекса работают по одному алгоритму. Специализация зон определяется только тем, откуда идут сигналы. В пользу этого принципа говорит довольно известный эксперимент, в котором подопытным новорожденным мышам хирургически изменили сенсорные пути так, чтобы они вели в другие сенсорные области. Мозг этих мышей приспособивался к тому, что они получают информацию с других сенсоров, и мыши вполне могли питаться, ориентироваться в пространстве.

3) Неокортекс большей частью работает во временной зоне и выявляет временные паттерны. Из-за этого принципа классическая нейронная сеть просто не может быть хорошей моделью, так как из-за отсутствия

обратных связей она не может распознавать последовательности сигналов (если только не упаковывать последовательности в один сигнал, например упаковка звука в изображение гистограммы, но такой подход нетипичен для мозга, поэтому в данном контексте он неприменим).

2.1 Биологические структуры и их модели

2.1.1 Нейрон

Обычно минимальной рассматриваемой структурой в работах по искусственному интеллекту и машинному обучению берут нейрон (рис. 2.3).

В наиболее простейших представлениях нейрон делят на 3 части - на сому (тело клетки), аксон и дендриты. Дендриты присоединяются синапсами к аксонам других клеток и таким образом получают сигналы, которые передаются в сому. Если в определенном месте сомы (аксоном холме) достигается какое-то критическое значение электрического потенциала, то сома активирует аксон, по которому сигнал передается в другие клетки.

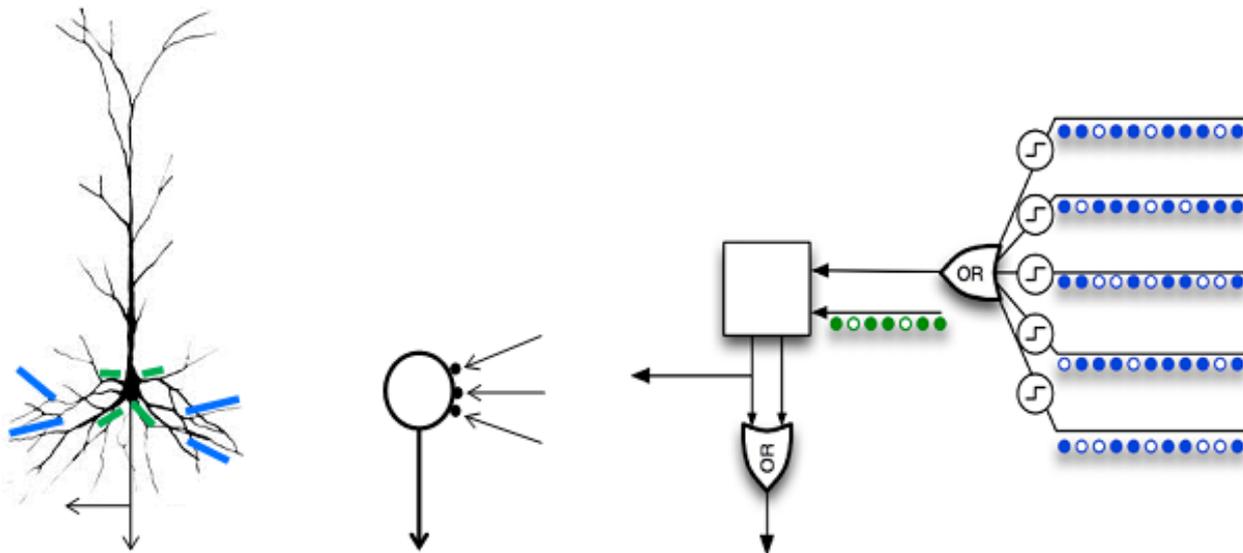


Рис. 2.3: На левом рисунке представлена схема реального нейрона, с геометрией дендритных деревьев, на среднем модель, используемая в классических сетях, справа модель используемая в НТМ. Следует заметить, что на левой схеме некоторые дендриты входят непосредственно в клетку, а некоторые образуют дендритные деревья

Именно такая модель используется в классических нейронных сетях. Однако исследования, проведенные в последние десятилетия показали, что

такая модель существенно отличается от происходящего в коре мозга. Дело в том, что дендриты также можно разделить на разные категории - те, которые вырастают напрямую в клетку, и те, которые образуют дендритные деревья. Те дендриты, которые вырастают в клетку (проксимальные дендриты) ведут себя примерно так же, как и в классической модели, однако дальние дендриты показали существенно нелинейное поведение, которое было названо дендритным спайком [12]. Это поведение дендритов меняет картину простого интегрирования сигналов, и заставляет склоняться к мысли, что в дендритах также проходит какая-то обработка информации. Подробнее об дендритных спайках можно прочитать в [13].

В разделе 3.4 будет показано как нейрон НТМ пытается моделировать такое поведение.

2.1.2 Колонка

Нейроны в коре головного мозга организуются в так называемые колонки. Впервые они были открыты командой Хьюбела и Визеля [10] (за что впоследствии получили нобелевскую премию). Они обнаружили структуры которые активировались при предъявлении определенных видов сигналов, а точнее на предъявление линий под определенным углом реагировали определенные колонки клеток (в этих экспериментах изучалась первичная визуальная кора).

Дальнейшие исследования показали, что подобная организация клеток наблюдается по всей коре головного мозга. При этом найдено много фактов, которые наводят на мысль, что колонки - минимальная структура, несущая семантический смысл, то есть каждая колонка реагирует на свой тип сигнала и становится активна (это значит, что подавляющее большинство клеток в колонке становятся активными). В первичной визуальной коре этим сигналом может быть кусок линии под определенным углом, в височной доле колонка может активироваться когда в поле зрения попадает лицо определенного человека.

Колончатая организация очень важна для работы мозга, и НТМ, в отличии от других моделей, также использует эту организацию.

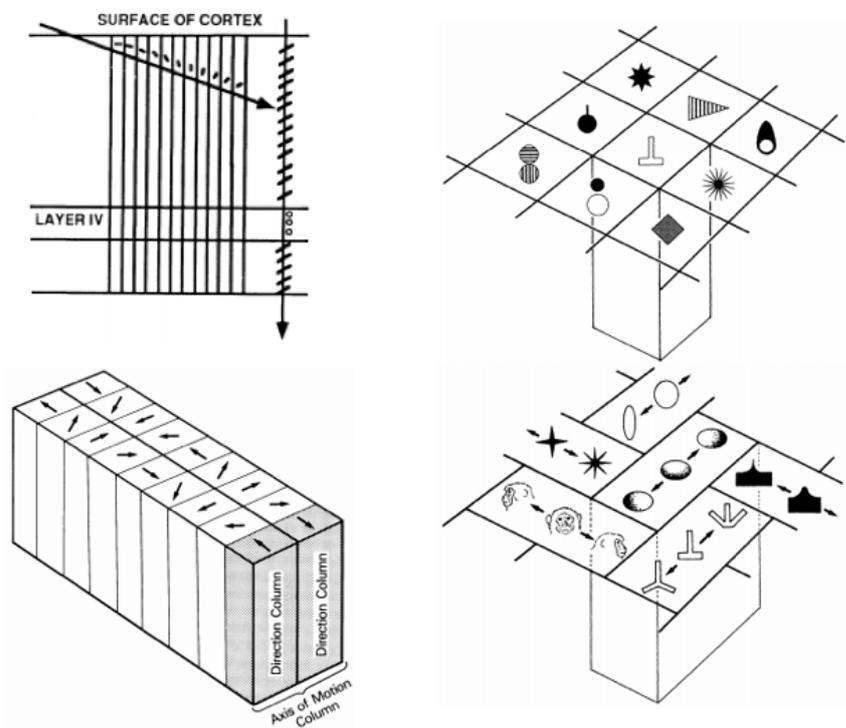


Рис. 2.4: Схематичное представление сигналов, на которые реагируют колонки

Глава 3

Описание модели

В этой главе будет представлена модель НТМ и обучающий алгоритм (Cortical Learning Algorithm, CLA). В главе 3.2 будет дано описание принципа хранения данных, называемый разреженные распределенные представления (Sparse Distributed Representation, SDR), которые используются в НТМ. В главе 3.3 будут описаны энкодеры для некоторых типов данных. Далее в главе 3.4 будет приведена подробная модель нейрона, используемая в НТМ. После этого будут описаны алгоритмы обучения НТМ, функционально разделенные на 2 части: пространственный группировщик (spatial pooler, SP, глава 3.5) и временной группировщик (temporal pooler, TP, глава 3.6)

3.1 Обзор НТМ

Как следует из названия модели (иерархическая темпоральная память), НТМ - это некая система памяти. Сама сеть состоит из иерархически организованных регионов. Понятие региона позаимствовано из нейробиологии и означает участок коры, обрабатывающий определенный тип информации. На рисунке 3.1 показан пример соединения четырех регионов в сеть. На нижний регион подается информация из сенсоров. После обработки информации на нижнем уровне, она передается на более высокий уровень. Стоит заметить, что информация проходит не строго снизу вверх (как в классических нейронных сетях), существуют информационные потоки внутри региона.

Чем выше слой сети - тем более абстрактная информация кодируется

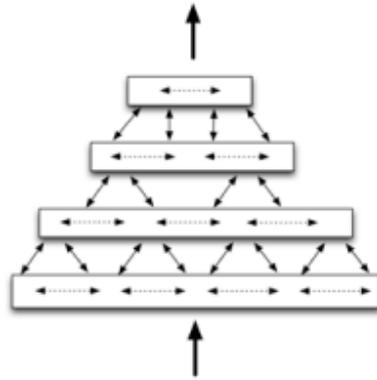


Рис. 3.1: Соединение четырех регионов в одну сеть

в регионах. Пример того, как это происходит в мозге был рассмотрен в главе 2.

На одном слое сети может быть несколько регионов. Такая архитектура сети имеет смысл, когда сеть получает сигналы с разных типов датчиков (рисунок 3.2), например аудио и видео сигналы. После обработки нескольких слоев, сигналы объединяются на высоких уровнях организации.

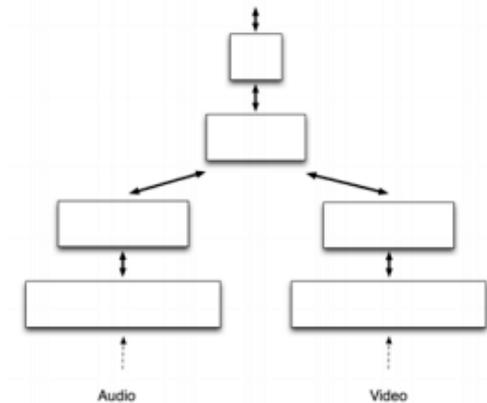


Рис. 3.2: Соединение регионов из разных сенсоров в одну сеть

В регионах находятся клетки, организованные в структуры кортикальных колонок (рис. 3.3). Клетки соединены как с предыдущем слоем, так и с другими клетками данного слоя. Подробнее о схеме соединения клеток и колонок друг с другом написано в главе 3.4.

В НТМ данные представляются в виде разреженных представлений, SDR (подробнее об этом представлении данных написано в главе 3.2). Для того, чтобы представить данные в формате SDR сначала данные с сенсоров преобразуют с помощью энкодеров в битовый массив, потом в помощью алгоритма пространственного группировщика (SP) в SDR.

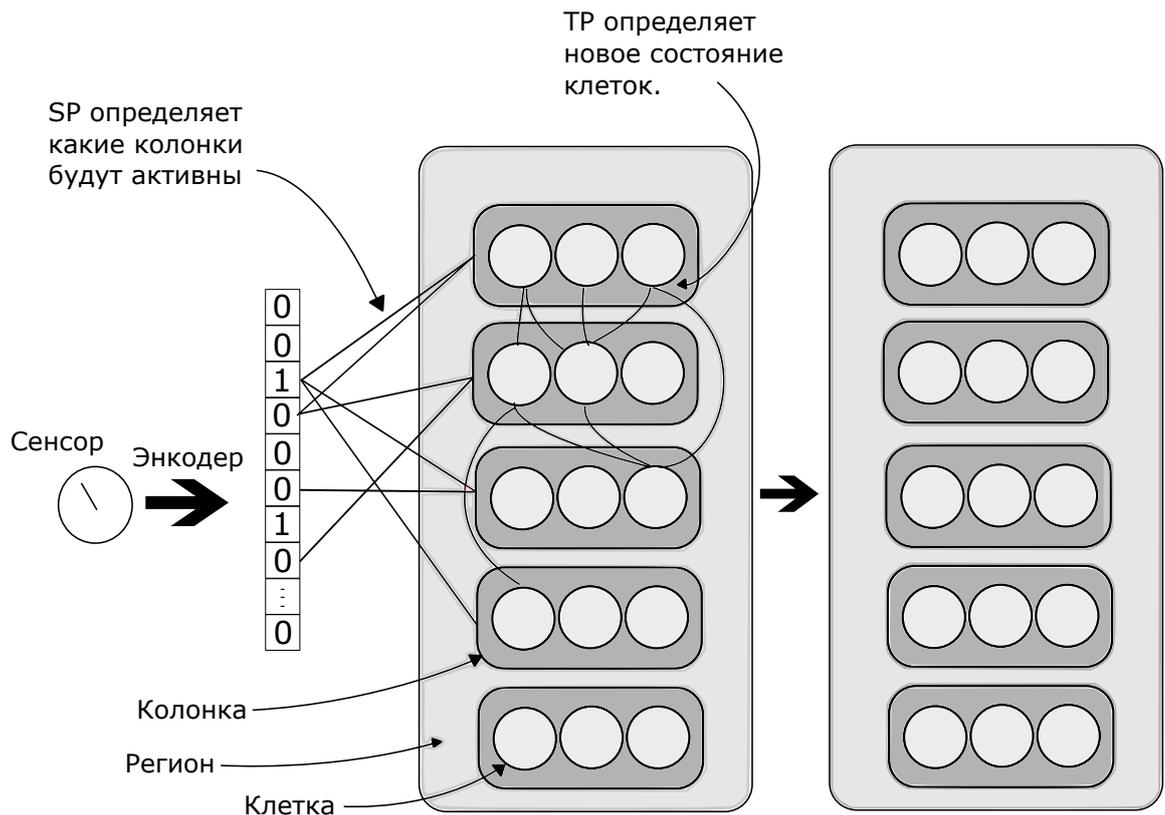


Рис. 3.3: Общая схема работы НТМ. Результат работы ТР далее используется как входные данные для региона следующего слоя

Основная задача SP – установить разреженность данных в заданных пределах (обычно это 2%, подробнее в главе 3.2). В терминах сети это значит, что SP выбирает какие колонки в регионе будут активными. Также SP может обучиться пространственным паттернам в данных.

После работы SP начинает работу алгоритм временного группировщика (ТР). Задача ТР - запоминать переходы между SDR (то есть запоминать последовательности активации колонок), правильно обновляя соединения между нейронами региона. Иными словами ТР пытается найти паттерны во временной области. Например сеть можно обучить на сигналах АВ, АС и АД (три последовательности, состоящие из двух сигналов), то при предъявлении А сеть должна выдать сигнал В+С+D (это общее представление сигнала, более подробно в главе 3.2). Но если сеть обучить на сигналах ABCD и XBCY, то при предъявлении ABC она должна выдать D, а не D+Y.

Результатом работы ТР является список нейронов и их новых состоя-

ний (результатом же работы SP является список состояний колонок). Этот список передается на следующий слой сети.

Именно благодаря TP и связям клеток внутри региона сеть является системой памяти, она может запоминать в себе ранее представленные последовательности образов.

В плане обучения SP модифицирует соединения между слоями, а TP соединения внутри слоя.

3.2 Пространственные разреженные представления

Пространственные разреженные представления (Sparse Distributed Representation, далее SDR) - формат данных для НТМ. Любой тип данных в этом формате представляются как бинарный массив с большим количеством битов, причем лишь немногие из этих битов активны (равны единице). Количество активных битов - это настраиваемый параметр сети, но обычно его ставят в районе 2% (вообще говоря понятие SDR выходит за рамки НТМ, как более общая концепция, но в терминах НТМ можно считать активный бит за активную колонку).

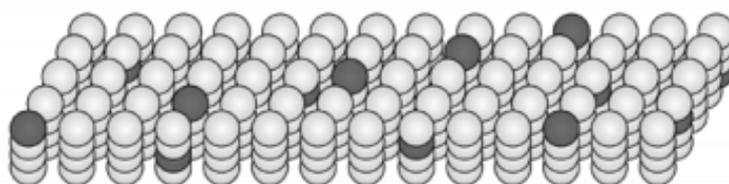


Рис. 3.4: Пример SDR. Несмотря на большое количество колонок и клеток, активны лишь несколько из них

Такой формат представления данных противопоставляется сжатому представлению, например таблице ASCII. В сжатом представлении используется как можно меньше бит, и обычно данные кодируются всеми возможными комбинациями нулей и единиц.

Одно из важнейших свойств SDR состоит в том, что каждый бит имеет семантическое значение, то есть когда данные переводят в формат SDR, то пытаются сделать так, чтобы семантически близкие данные имели

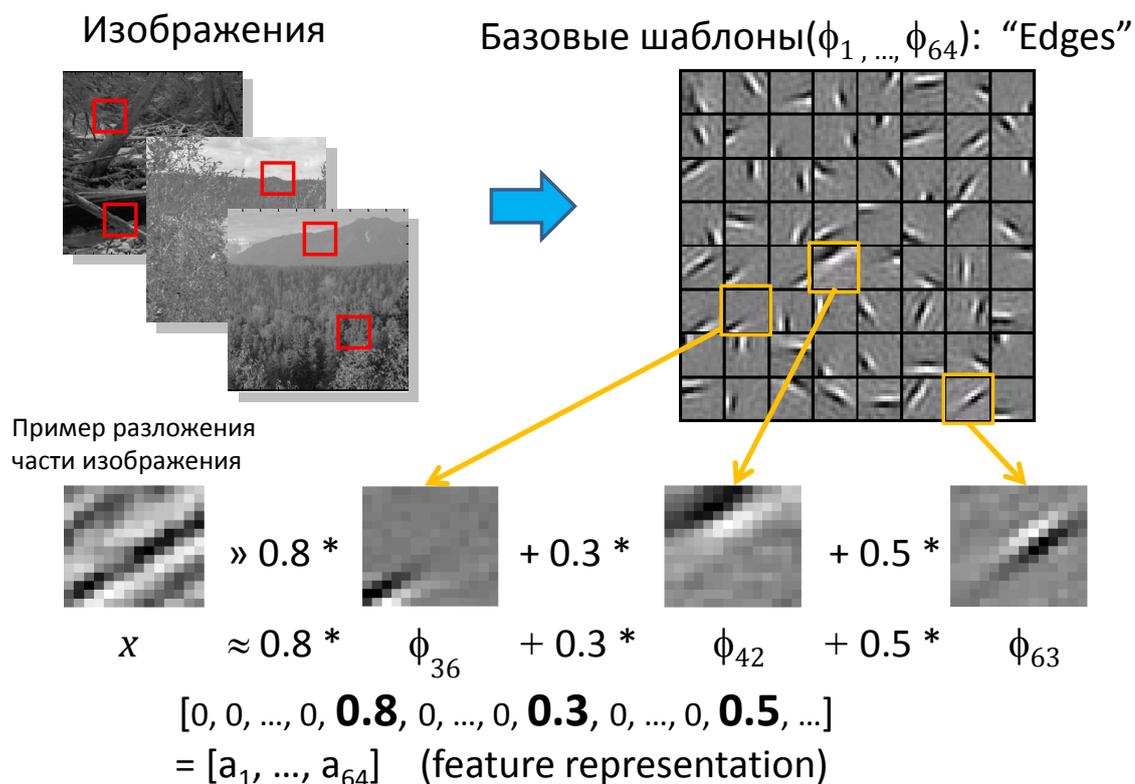


Рис. 3.7: Пример перевода пиксельного представления изображения в разреженное представление признаков (feature representation). Такая операция происходит внутри сверточных нейронных сетей.

На рис 3.7 можно увидеть как работают первые уровни сверточной нейронной сети. Обучая сеть, можно получить набор базовых шаблонов, которыми можно представлять изображение. То есть вместо набора пикселей сверточная сеть представляет каждое маленькое окно в изображении как разреженный вектор, в котором указано какие из базовых шаблонов присутствуют в этом окне.

Именно благодаря такому представлению данных сверточные сети в свое время произвели переворот в задачах распознавания изображений, ведь в отличие от пиксельного представления, где мы имеем просто набор яркостей пикселей, вектора признаков имеют какие-то семантические значения, они говорят какие формы содержатся в изображении.

Эти базовые шаблоны тоже можно представить иерархией, где на нижних уровнях изображение представляется как набор линий, а на высоких уже как более абстрактные объекты, вроде глаз, рта, лица в сети

обученной на распознавание лиц (рис 3.8).

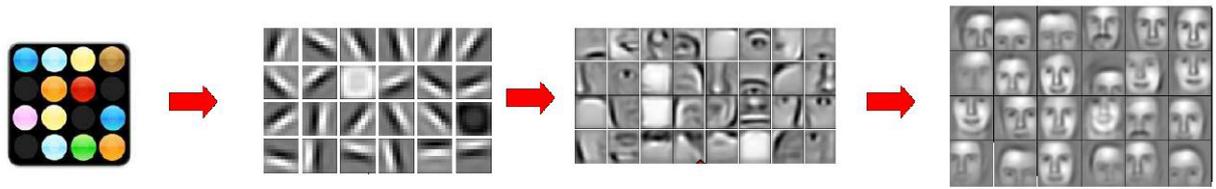


Рис. 3.8: Иерархия признаков в сверточных нейронных сетях, обученных на распознавание лиц.

Также аналогию SDR представления можно увидеть в появившейся недавно технике обучения нейронных сетей, называемой dropout. Согласно ей при обучении сверточных нейронных сетей нейроны с некоторой вероятностью "отключаются" и не участвуют в обучении (для конкретной точки из набора данных). Эта техника позволяет лучше справляться с переобучением моделей, делая представление сети более разреженным [16].

3.3 Энкодеры

Прежде чем подавать данные на вход НТМ их нужно представить в пространственном представлении (DR), то есть в виде бинарного массива, имеющего следующие свойства:

- 1 Семантически близкие данные переходят в представления с общими активными битами
- 2 Один и тот же сигнал всегда должен получать одно и то же представление
- 3 Полученное представление должно иметь одну размерность для любых входов

Для перевода сигналов в это представление используются энкодеры. Нужно заметить, что энкодеры не обязаны давать SDR, то есть число активных бит не обязательно быть равным какому-то определенному, наперед заданному числу. Переводом такого представления в SDR занимается SP. Однако, если энкодер сможет сразу представлять данные в SDR формате (или хотя бы близкой разреженности), это сильно ускорит работу сети.

В этом разделе мы рассмотрим несколько примеров энкодеров.

3.3.1 Числовой энкодер

Подход к реализации этого энкодера позаимствован у кохлеарных волосков (рис 3.9) – специальных клеток, которые резонируют с определенной частотой и, при резонанции, активируют подсоединенные к ним нейроны. Однако при прослушивании звука частотой, например, 50Гц будет активен не только нейрон, непосредственно подсоединенный к волоску с резонансной частотой 50Гц, но и соседние. Так наше тело кодирует сигналы с похожими частотами похожим образом.

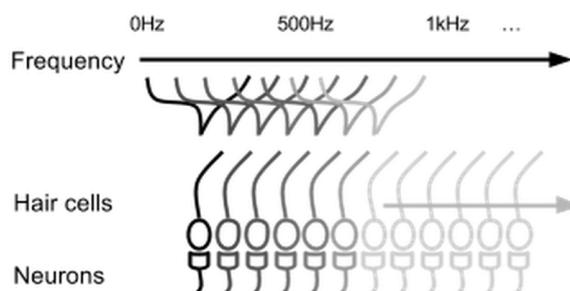


Рис. 3.9: Кодирование частоты звука в ушах.

Так же можно кодировать произвольные числа в диапазоне. Например кодирование чисел от 1 до 10 может выглядеть таким образом.

01 – **111**000000000

02 – 0**111**00000000

04 – 000**111**000000

10 – 000000000**111**

Общее количество бит представления, а также число активных бит - настраиваемые параметры.

3.3.2 Энкодер категорий

Иногда нужно закодировать какие-то непересекающиеся категории. Тогда их нужно кодировать так же как и числа, но без пересекающихся активных битов.

день – 1111 0000

ночь – 0000 1111

3.3.3 Временной энкодер

Временной энкодер похож на числовой. Единственное отличие в том, что иногда нам нужен будет циклический энкодер, например для часов. Если в сеть будет передаваться информация о часе дня, то будет необходимо, чтобы 3 часа и 4 были также похожи, как 23 часа и полночь. Для этого нужно зациклить выставление активных бит, как на рисунках 3.10, 3.11

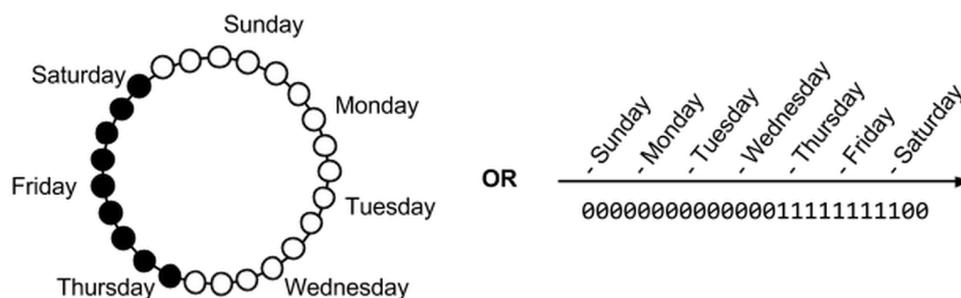


Рис. 3.10: Пример циклического энкодера.

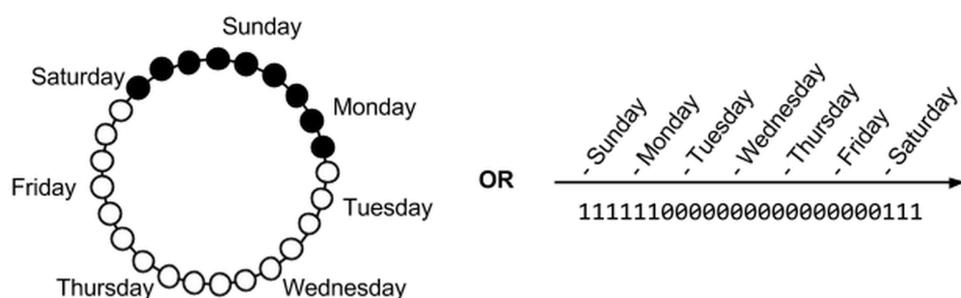


Рис. 3.11: Пример циклического энкодера.

3.3.4 Геопространственный энкодер

Геопространственные данные (geospatial data) - данные GPS координат. Энкодер для таких данных должен обладать двумя свойствами. Во-первых близкие друг к другу координаты должны иметь семантическое сходство, поэтому у них должны пересекаться активные биты. Во-вторых разрешение такого энкодера должно зависеть от скорости объекта (ведь для автомобиля на трассе точки на расстоянии 200 метров довольно близки, а значит похожи, для пешехода это уже существенное расстояние).

Для этого карта будет разбита на малые сектора, каждый сектор будет представлен в SDR. От текущего местонахождения будет взят радиус, зависящий от скорости (чем больше скорость, тем больше радиус). В SDR будут добавляться только N участков карты внутри радиуса (число N выбирается так, чтобы достигнуть желаемой разреженности). Чтобы выбрать N клеток нужно каждой клетке присвоить случайное значение между 0 и 1, далее выбрать N клеток с самыми большими числами (рис. 3.12).

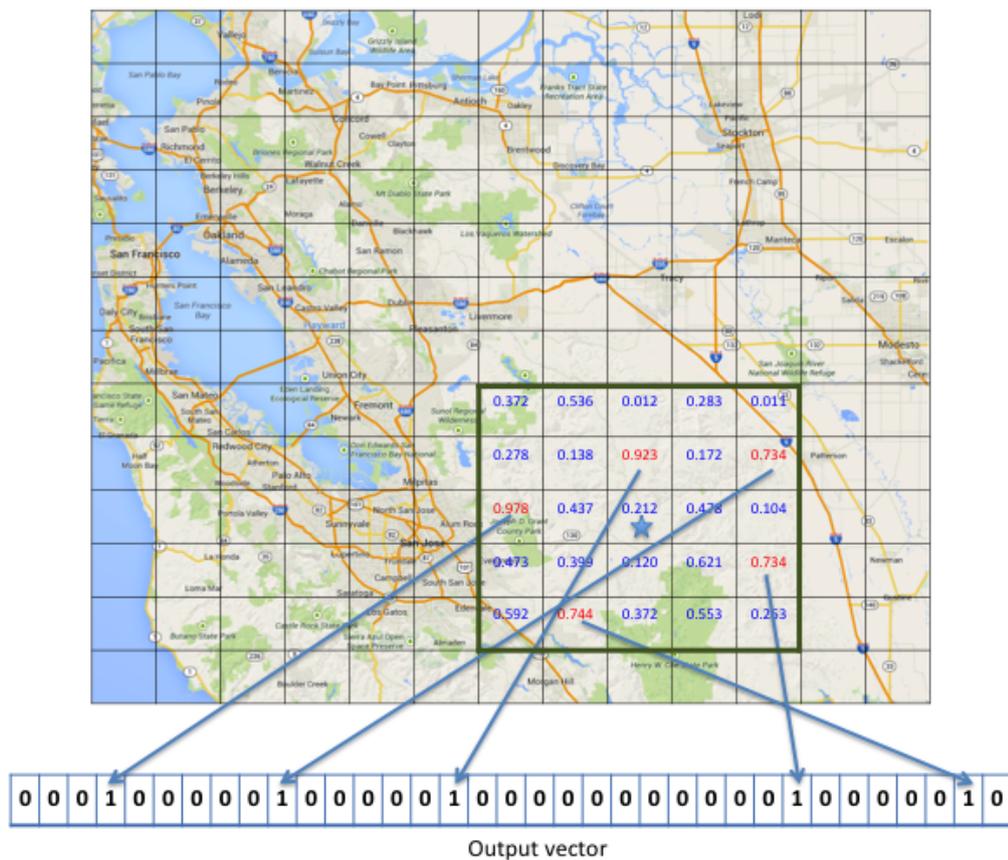


Рис. 3.12: Выбор N значимых квадратов из радиуса

Таким образом, достигается первое свойство: у близких точек будут совместные активные биты (рис. 3.13).

Также это позволяет получить второе свойство. На рис. 3.14 видно, что хоть перемещение точки и больше, но из-за большего радиуса все-равно есть пересекающиеся активные биты.

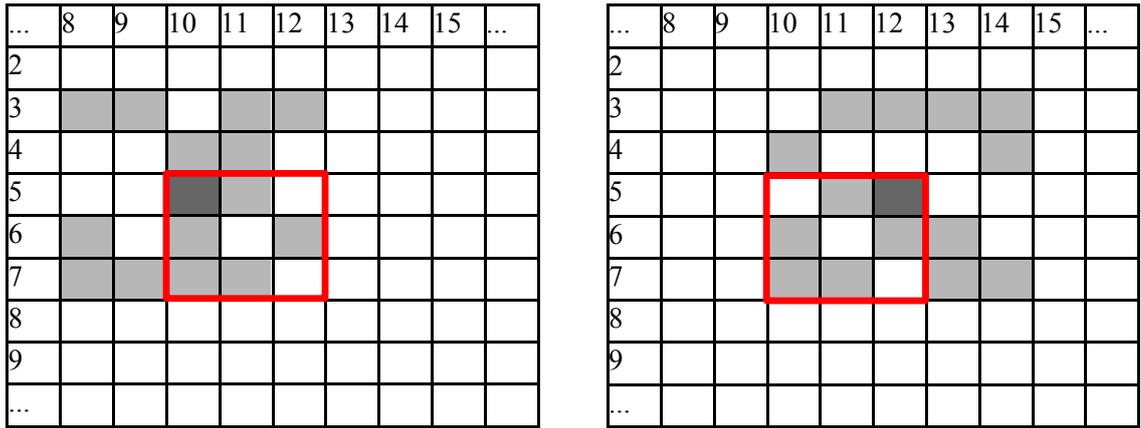


Рис. 3.13: Показанно перемещение объекта и N выбранных клеток в радиусе. Как видно из рисунка, часть клеток для двух точек одинаковая.

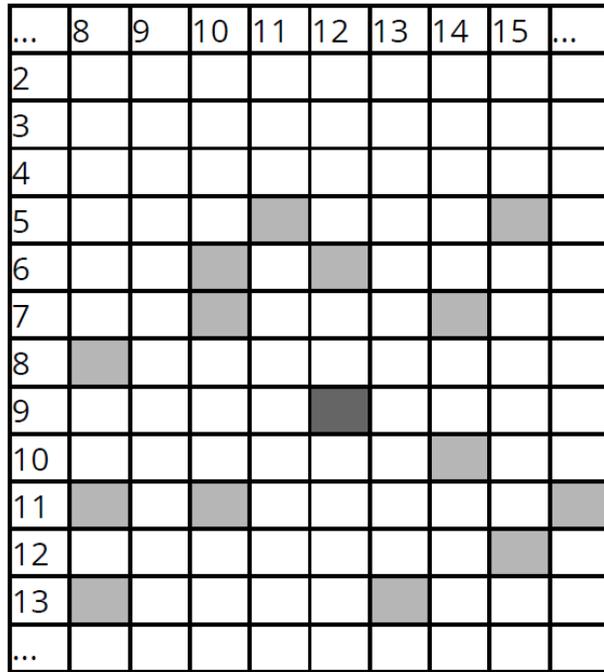


Рис. 3.14: Пример энкодера с большей скоростью

3.4 Клетка НТМ

Как уже говорилось в разделе 2, модель нейрона в НТМ отличается от модели нейрона в классических нейронных сетях. В персептроне нейрон берет взвешенную сумму всех своих входов, проводит через функцию активации, и полученное число считается выходом нейрона (рис. 3.15)

В НТМ модель нейрона (которая в терминах этой теории называется просто клетка) дискретна и может пребывать в трех состояниях: неактивна,

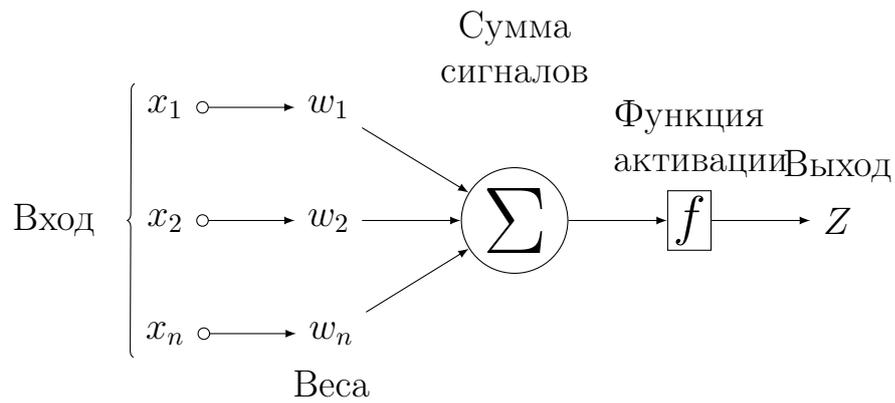


Рис. 3.15: Модель искусственного нейрона

активна, состояние предсказания (англ. predictive state).

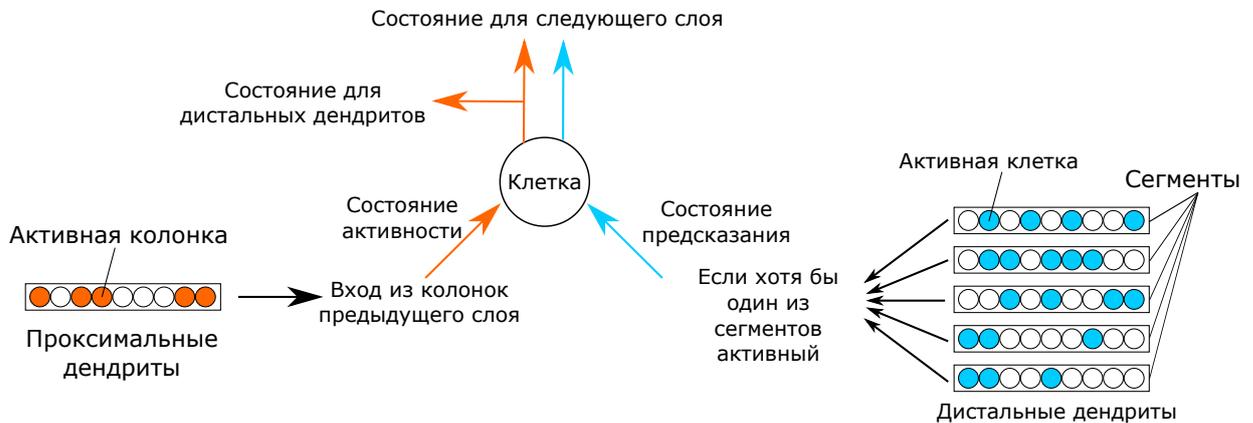


Рис. 3.16: Клетка НТМ

В отличие от классических нейронов, в клетках НТМ есть 2 вида входов, так называемые проксимальные (ближние) дендриты и дистальные (удаленные) дендриты.

Проксимальные дендриты соединяют биты входящего сигнала (с сенсоров или с предыдущего слоя) со всеми нейронами колонки посредством синапсов. Активность синапсов суммируется и, если сумма больше заданного порога, клетка переходит в активное состояние. Каждая клетка колонки соединена с одинаковыми входными битами, но разными синапсами.

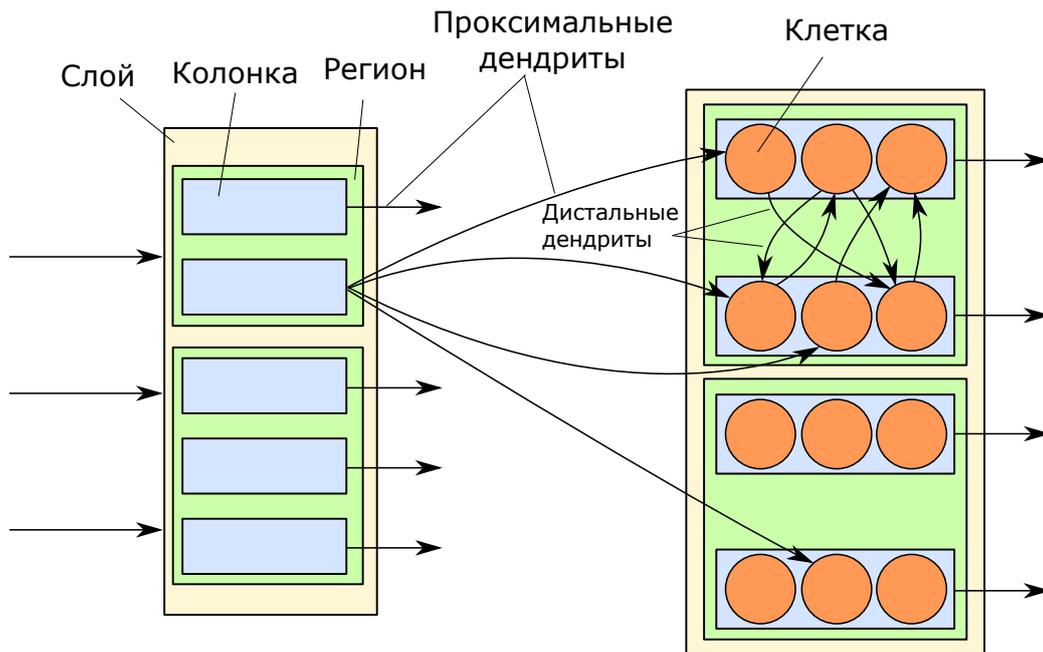


Рис. 3.17: 2 региона НТМ. Показанны как дендриты соединяют клетки

Дистальные дендриты соединяют клетки внутри одного региона. В каждой клетке есть список так называемых сегментов, каждый сегмент состоит из соединений с определенными клетками, посредством дистальных дендритов. Если в одном из сегментов суммарная активность больше заданного порога, то клетка переходит в состояние предсказания (если до этого была в состоянии неактивности). Сегмент дистальных дендритов запоминает нужное ему состояние региона НТМ путем формирования связей с его клетками, которые ранее были активны в один и тот же момент времени. Таким образом, сегмент запоминает состояние региона, которое предшествовало активации его клетки проксимальными дендритами. Далее сегмент просто следит за состоянием этих «своих» клеток региона НТМ, активация которых предсказывает, что и его клетка станет сейчас активной.

В данном случае сегмент - аналог ветви дендритного дерева. Иначе говоря, этим механизмом моделируется дендритный спайк.

3.5 Пространственный группировщик

Главная задача пространственного группировщика (Spatial Pooler, далее SP) - правильно активировать колонки, для получения разреженного представления.

Каждой колонке региона сопоставлены биты входящего сигнала (так называемое рецептивное поле, которое не меняется при работе сети). Каждому такому соединению колонка-бит сопоставляется синапс с весом соединения (число в промежутке от нуля до единицы, которое меняется в ходе обучения). Если входящий бит активен (равен единице) и вес синапса больше заданного порога, то вход считается соединенным (connected в оригинальных статьях) с колонкой. С чем большим количеством входов соединена колонка, тем больше её шансы стать активной.

SP преследует несколько целей.

- **Использовать все колонки.** Ситуации, где какая-то часть колонок никогда не бывает активна нежелательна. Поэтому SP отслеживает колонки, которые редко активируются (по сравнению со своими соседями) и усиливает их синапсы.
- **Поддержание разреженности.** Данные, поступающие в SP, зачастую содержат слишком много активных битов. Именно SP гарантирует заданный процент активных колонок.
- **Избегание тривиальных паттернов.** Чтобы не было ситуации, когда SP просто напрямую передает данные TP, никак не обрабатывая, в SP существует минимальный порог активных дендритов. Например, если стоит порог 10, то чтобы активировать колонку, должны быть активны не менее 10 входов (даже если коэффициенты синапсов велики).
- **Избегание лишних соединений.** Обратная ситуация, когда колонка соединена со слишком большим количеством входных битов также нежелательна.

Для того, чтобы не было большого числа активных колонок, задается параметр радиуса ингибиции (inhibition radius). SP следит за тем, чтобы

в радиусе ингибиции колонки было активно не более N соседних колонок (число N выбирается так, чтобы во всём регионе было активно заданный процент колонок). Обучение SP производится изменением весов синапсов.

Приведем формальный алгоритм SP.

Инициализация. SP

- 1 Задается рецептивное поле.
- 2 Для каждого входного бита случайно задается коэффициент силы синапса.
- 3 Задается радиус ингибиции.

Вычисление SP. Для каждой колонки

- 1 Вычисляется сколько входов соединены с колонкой. Если суммарная активность синапсов выше порога, то колонка становится кандидатом на активацию.
- 2 Проводится ингибирование колонок, в радиусе ингибиции остается только N активных колонок.
- 3 Выполняется обучение по Хеббу. Если вход колонки активен, то синапсы этого соединения усиливаются, если вход неактивен, то ослабляются.
- 4 Усиление синапсов колонок, которые получают мало входа.
- 5 Обновление радиуса ингибиции.

Если обучение не нужно, то останавливаемся на 2 шаге. После SP у нас появляется информация какие из входных битов соединены с какими колонками. Эта информация далее используется в TP.

3.6 Темпоральный группировщик

После работы SP мы получили список активных колонок, теперь мы должны получить список активных клеток в этих колонках.

НТМ обучают на последовательностях конечных длин. Как сеть понимает, что закончилась одна последовательность и началась другая? Это можно сделать тремя способами. Во-первых можно вручную сообщать сети, что началась новая последовательность (вызовом определенной функции). Во-вторых, если вручную сделать это невозможно, то сеть будет считать, что началась новая последовательность, если активируются слишком много клеток, которых ранее не были в состоянии предсказания. В-третьих, если длина сигнала превышает какого-то фиксированного значения. Для хорошей работы сети крайне важно правильно устанавливать начало новой последовательности.

Для начала рассмотрим как ТР рассчитывает **состояния клеток**.

1 Активируем клетки

- 1.1. Если начата новая последовательность, активируем начальную клетку (первую, или случайную) в каждой активной колонке
- 1.2. Иначе в каждой активной колонке активируем клетки в состоянии предсказания
- 1.3. Если в колонке нет клеток в состоянии предсказания, то активируем все клетки в этой колонке

2 Смотрим на активность сегмента в каждой клетке. Если к клетке подключено достаточно активных клеток и сама клетка не активна, то переводим эту клетку в состояние предсказания.

3 Если клетка на этом шаге не перешла в состояние активности или предсказания, то клетка переходит (или остается) в состоянии неактивности.

После этих действий у нас уже имеется выход ТР, если мы не хотим проводить обучение, то здесь можно остановиться.

Обучение ТР.

- 1 Для каждого активного сегмента активной клетки обновляем синапсы каждого соединения в сегменте по правилу Хебба (усиливаем синапсы с активными и ослабляем с неактивными клетками), но в отдельную память (считаем эти изменения временными).

- 2 Если клетка перешла в активное состояние из неактивного, то применяем временные изменения сегментов.
- 3 Если клетка перешла в активное состояние из состояния предсказания, то временные изменения сегментов не применяются.

3.7 Поиск аномалий

Будем считать аномалией количество клеток, которые были в состоянии предсказания, но не перешли в активное состояние. На основе этого получим формулу расчета коэффициента аномалии:

$$anomalyScore = \frac{|A_t| - |(P_{t-1} \cap A_t)|}{|A_t|},$$

где A_t – Активные клетки в момент времени t , P_t – клетки в состоянии предсказания в момент времени t , $|A|$ – количество элементов в множестве A .

Глава 4

Проведение эксперимента

В этой главе будут описаны детали эксперимента с НТМ.

4.1 Входные данные

GPS данные записывались телефоном Highscreen Alpha Rage приложением GPSLogger. Из данных сохранялись только значения широты, долготы, скорости и времени. Время записи не являются данными для обучения, они использовались только для правильной сортировки точек в маршрутах и для удобства работы с данными.

Так как в данных были очевидные выбросы точек (видимо в связи с плохой связью со спутниками), они сглаживались медианным фильтром.

4.2 Конфигурация НТМ

Как и при использовании нейронных сетей, в НТМ исследователю нужно самостоятельно выбирать многие параметры, такие как количество слоев, колонок, клеток в колонках и т. п. На сегодняшний день нет стройной теории, помогающей выбирать эти параметры и исследователю нужно руководствоваться работами своих коллег, своим чутьем, а также результатами экспериментов. Иными словами нужно пробовать обучать сеть с разными параметрами и выбирать те, которые хорошо работают с данной задачей. Это влечет за собой то, что выбранные конфигурации возможно не идеальны, и можно получить более лучшие результаты с более тонкой

настройкой.

Параметры НТМ, взятые в эксперименте приведены в таб. 4.1. Так как в решаемой задаче присутствовали данные только одного типа, была взята однослойная НТМ с одним регионом.

Тип энкодера	GeospatialCoordinateEncoder
Чувствительность энкодера	5м
Количество бит энкодера	2048
Количество активных бит	51
Количество регионов	1
Количество слоев	1
Количество колонок	2048
Количество клеток в колонке	32
Максимальное кол. синапсов в сегменте	32

Таблица 4.1: Параметры сети.

4.3 Оборудование и программное обеспечение

Все эксперименты проводились на машине с процессором Intel Core i7-3610QM (2.3 GHz). Использовалась реализация НТМ компании Numenta под названием NuPIC. Вспомогательные скрипты и модификации пакета проводились на языке Python. Для визуализации результата работы сети были использованы карты Google Maps и скрипты на языке javascript.

4.4 Результаты эксперимента

На рисунке 4.1 видно несколько проходов одного маршрута в разное время. Цветом обозначено значение коэффициента аномальности для каждой точки, от зеленого цвета (аномальность 0) до красного (аномальность 1).

На рисунке а) этот маршрут встретился сети в первый раз, поэтому каждая точка отмечена аномальной. При каждом проходе сеть более лучше запоминала этот маршрут, и значения аномальности уменьшались.

На рисунке 4.2 показан случай при котором была пройдена часть предыдущего маршрута, но далее был выбран другой путь.

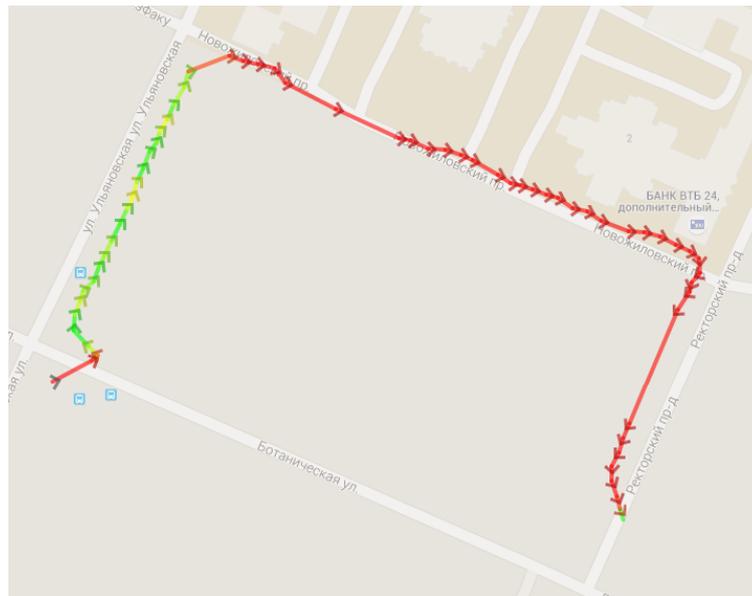


Рис. 4.2: Случай, когда часть маршрута оставалась прежней, а часть поменялась.

На рисунке 4.3 показан случай, при котором маршрут оставался таким же, но пройден в другую сторону.

Так же были проведены эксперименты при котором сеть обучалась на множестве других маршрутов (было смоделировано 100 маршрутов). При этом первоначальный маршрут не был "забыт" сетью, она реагировала на него как и прежде.

Заключение

В данной работе рассматривалась модель иерархической темпоральной памяти (НТМ). Это система памяти, созданная под влиянием работы неокортекса млекопитающих, но в отличие от остальных моделей нейронных сетей, используемых в задачах машинного обучения, эта модель более точно отображает процессы происходящие в мозге. Целью данной работы являлась оценка применимости данной модели в задачах машинного обучения, несмотря на то, что при составлении модели не использовались обычные для таких моделей статистические приемы, не имеющие аналогов в мозге. В качестве конкретной задачи было выбрано поиск аномалий в GPS данных.

В главах 3-4 было подробно рассмотрена модель НТМ и алгоритмы обучения CLA. Было показано, что наибольший потенциал данная модель раскрывает на задачах, в которых нужно искать сложные временные паттерны. Это достигается за счет разбиения клеток сети на группы, называемые колонками, и взаимодействием между клетками одного слоя.

Было показано, что НТМ успешно справляется с поставленной задачей. Сеть быстро обучается находить предъявленные ранее маршруты и отмечать не представленные ранее точки пути (аномалии маршрута). Также было показано, что сеть запоминает не просто набор точек, а их последовательность и, например, тот же маршрут, но пройденный в обратную сторону будет отмечен как аномальный.

В ситуации предъявления большого количества маршрутов, из сети не пропадает информация о маршрутах, предъявленные её давно. Иными словами сеть надежно хранит информацию внутри себя.

Одним из недостатком модели является её молодость. Впервые модель НТМ была представлена в 2004 году, а алгоритмы её обучения CLA в 2009. Это создает некоторые трудности при изучении модели по ряду

причин. Первая причина – малое количество статей. Вторая причина состоит в том, что модель активно меняется, модифицируется, из-за этого довольно сложно отслеживать актуальное описание модели и алгоритмов. Третья причина – малое количество реализаций модели в виде программного обеспечения. Актуальная реализация NuPIC работает только на процессоре. Без реализации на видеокарте довольно сложно обучать модель на большом объеме данных.

Если говорить в целом о применимости НТМ в задачах машинного обучения, то модель однозначно имеет свою нишу в задачах поиска нетривиальных паттернов во временной области. В данной работе не рассматривались задачи другого типа, однако, судя по имеющимся публикациям, НТМ несколько уступает другим моделям (имеются в виду задачи обучения с учителем и поиск паттернов в пространственных данных).

Возможные дальнейшие исследования модели.

- 1 Добавить слои сети, что может помочь выявлять более высокоуровневые паттерны поведения.
- 2 Использовать время как вход с сенсора. То есть добавить ещё один регион, который будет принимать сигнал времени и объединить сигналы с двух регионов на следующем слое сети. В данный момент время движения никак не учитывается, а ведь прохождение обычного маршрута в необычный час тоже может считаться аномалией.
- 3 Обучение сети на большом количестве данных, собранных с нескольких объектов. Объединяя несколько сетей в одну (так как данная модель легко расширяется, это не представляет большой проблемы), можно попытаться искать аномалии в коллективном поведении.

Литература

1. Hawkins J., Blakeslee S. On Intelligence. Times Books, 2004. ISBN: 0805074562.
2. Chandola V., Banerjee A., Kumar V. Anomaly Detection: A Survey // ACM Comput. Surv. 2009. — Jul.. Vol. 41, no. 3. P. 15:1–15:58.
3. Augusteijn M. F., Folkert B. A. Neural network classification and novelty detection // International Journal of Remote Sensing. 2002. Vol. 23, no. 14. P. 2891–2902.
4. Frank R. J., Davey N., Hunt S. P. Time Series Prediction and Neural Networks // J. Intell. Robotics Syst. 2001. — May.. Vol. 31, no. 1-3. P. 91–103.
5. Hofmann D. B. T., Baker L. D., Hofmann T. et al. A Hierarchical Probabilistic Model for Novelty Detection in Text. 1999.
6. Ramaswamy S., Rastogi R., Shim K. Efficient Algorithms for Mining Outliers from Large Data Sets // SIGMOD Rec. 2000. — May.. Vol. 29, no. 2. P. 427–438.
7. Emamian V., Kaveh M., Tewfik A. Robust clustering of acoustic emission signals using the Kohonen network // Acoustics, Speech, and Signal Processing, IEEE International Conference on. 2000. Vol. 6. P. 3891–3894.
8. Gers F. Long Short-Term Memory in Recurrent Neural Networks. 2001.
9. Glorot X., Bengio Y. Understanding the difficulty of training deep feed-forward neural networks // JMLR W&CP: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics (AISTATS 2010). Vol. 9. 2010. — May.. P. 249–256.

10. D.H. Hubel, T.N. Wiesel. Receptive fields and functional architecture of monkey striate cortex // *Physiol.* 1968. Vol. 195, no. 1. P. 215–243.
11. Sherman S. M., Guillery R. W. The role of the thalamus in the flow of information to the cortex // *Philosophical Transactions of the Royal Society B: Biological Sciences.* 2002. Vol. 357, no. 1428. P. 1695–1708. <http://rstb.royalsocietypublishing.org/content/357/1428/1695.full.pdf>.
12. Hawkins J., Ahmad S. Why Neurons Have Thousands of Synapses, A Theory of Sequence Memory in Neocortex // *Frontiers in Neural Circuits.* 2016. Vol. 10, no. 23.
13. Hausser M., Spruston N., Stuart G. J. Diversity and Dynamics of Dendritic Signaling. // *Science.* 2000. Vol. 290. P. 739.
14. Ahmad S., Hawkins J. Properties of Sparse Distributed Representations and their Application to Hierarchical Temporal Memory // *CoRR.* 2015. Vol. abs/1503.07469.
15. Lee H., Battle A., Raina R., Ng A. Y. Efficient sparse coding algorithms // In *NIPS.* NIPS, 2007. P. 801–808.
16. Srivastava N., Hinton G., Krizhevsky A. et al. Dropout: A Simple Way to Prevent Neural Networks from Overfitting // *Journal of Machine Learning Research.* 2014. Vol. 15. P. 1929–1958.

Используемые аббревиатуры.

HTM – Hierarchical Temporal Memory, иерархическая темпоральная память.

CLA – Cortical Learning Algorithms, кортикальные алгоритмы обучения.

SDR – Sparse Distributed Representation, пространственные разреженные представления.

SP – Spatial Pooler, пространственный группировщик.

TP – Temporal Pooler, временной группировщик.