

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

**МОСКОВСКИЙ ФИЗИКО - ТЕХНИЧЕСКИЙ ИНСТИТУТ
(государственный университет)**

На правах рукописи

Нгуен Нгок Зиен

**АЛГОРИТМЫ ПОСТРОЕНИЯ АДАПТИВНОГО
ЯЗЫКОВОГО ЧЕЛОВЕКО-МАШИННОГО
ИНТЕРФЕЙСА ДЛЯ ПРОГРАММНЫХ СИСТЕМ**

Специальность 05.13.11 – «Математическое и программное обеспечение
вычислительных машин, комплексов и компьютерных сетей»

ДИССЕРТАЦИЯ

на соискание учёной степени
кандидата физико-математических наук

Научный руководитель:
доктор физико-математических наук, профессор
А.А. Жданов

Москва – 2018

Содержание

Список сокращений	4
ВВЕДЕНИЕ	5
ГЛАВА 1. Описание адаптивных нейроподобных программных элементов и конструкций, порождающих языковой человеко-машинный интерфейс в программной системе на примере системы автономного адаптивного управления	26
1.1. Описание элементов подсистемы ФРО метода ААУ	26
1.2. Описание структуры нейроподобных фильтров.....	33
1.3. Описание самоорганизующейся системы формирования и распознавания образов ФРО	42
1.4. Языковые явления и основная нейроподобная конструкция, порождающая язык в нейроподобной системе ААУ	48
1.5. Описание модели формирования базы знания на основе нейроподобных сетей	51
1.6. Выводы по главе 1	60
ГЛАВА 2. Алгоритмы построения самообучаемой системы распознавания статических образов и системы распознавания звуков на основе адаптивных нейроподобных элементов (нейронов)	61
2.1. Алгоритмы динамического построения сети для формирования и распознавания статических образов.....	62
2.2. Построение алгоритма самообучаемой нейроподобной сети для системы распознавания звуков	72
2.3. Выводы по главе 2.....	83

ГЛАВА 3. Реализация основных программных модулей адаптивной программной системы со свойствами языкового интерфейса на примере системы автономного адаптивного управления	85
3.1. Реализация ядра системы управления ААУ	85
3.2. Протокол обмена данными между модулями системы управления ААУ	87
3.3. Реализация модулей модификации структур нейроподобных сети	89
3.4. Реализация аппарата эмоций метода ААУ	93
3.5. Выводы по главе 3	94
ГЛАВА 4. Прикладная система автономного адаптивного управления с языковым человеко-машинным интерфейсом для БПЛА	95
4.1. Модель виртуальной среды для моделирования полета БПЛА на UNITY3D.....	95
4.2. Постановка задачи управления БПЛА с голосовым интерфейсом	98
4.3. Алгоритм функционирования системы управления БПЛА	100
4.4. Результаты экспериментов	108
Некоторые полезные приложения нейроподобных сетей из метода ААУ ...	119
СПИСОК ЛИТЕРАТУРЫ.....	126

Список сокращений

ААУ – автономное адаптивное управление

БЗ – база знаний

БПЛА - Беспилотный летательный аппарат

ДС – дерево событий

ИИ – искусственный интеллект

ИНС – искусственная нейронная сеть

ОУ – объект управления

ПИИ – подчиненный искусственный интеллект

УС – управляющая система

ФРО – формирование и распознавание образов

ВВЕДЕНИЕ

Проблема создания программных систем с языковым интерфейсом

Настоящее время характеризуется появлением и быстрым развитием производства программно-насыщенных систем, требующих человеко-машинных языковых интерфейсов на основе естественного языка. Растут затраты на разработку и развитие человеко-машинных интерфейсов в жизненном цикле программного обеспечения. Усилия разработчиков направлены на объединение различных форм общения человека с компьютером. Одновременно стараются обеспечить персонализацию человеко-машинных интерфейсов, не только за счет априорной или опциональной настройки параметров интерфейса под индивидуальные особенности пользователя, но и за счет автоматической адаптации интерфейса к особенностям человека, в частности – его речевой, языковой активности. В связи с этим высокую актуальность приобретают программные системы, предназначенные для работы с естественным языком – системы распознавания речи, синтеза речи, и т.п. Благодаря таким интеллектуальным человеко-машинным интерфейсам возможно создание различных программных сред, как рабочих, так и игровых.

Сегодня система распознавания голоса разрабатывается и развивается многими крупными компаниями по всему миру. Качество распознавания голоса становится все более высоким. В сочетании с моделью естественного языка, много программных обеспечений создано на основе этих систем: голосовые помощники, перевод речи в текст и вопросно-ответные системы. Так же в системах управления все большую роль отводят языковому аспекту. Настоящая работа также связана с проблемой языка.

Языковая система – множество элементов языка, связанных друг с другом теми или иными отношениями, образующее определённое единство и целостность [10,27,28]. Язык – это знаковая система естественно или

искусственно созданная и соотносящая понятийное содержание и типовое звучание [10,27,28]. В человеческом обществе язык является главным средством передачи мысли. Благодаря языку люди начали обмениваться опытом, планировать свои совместные действия. Поэтому, по нашему мнению, язык может обеспечить естественным способом для пользователей выполнять такие задачи, как вводные команды через человеко-машинные интерфейсы или передача знаний через машинно-машинный интерфейс с использованием языковых идентификаторов.

В 1966 году была создана программа ELIZA [6], которая могла общаться с человеком. Эта программа представляет собой большой набор правил, по которому она находила определенные паттерны в речи человека. В ответ на наличие соответствующего паттерна входной информации, ELIZA определенным образом трансформировала его и выдавала информацию на выход (рис. 1).

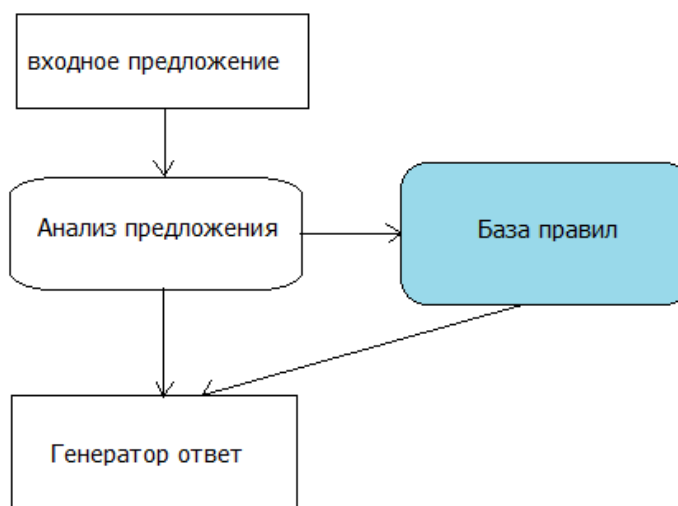


Рис. 1. Алгоритм чат-бота ELIZA.

ALICE (Artificial Linguistic Internet Computer Entity) – разработанная в 1995 году программа с искусственным интеллектом. Это первая программа чат-бота смогла обрабатывать естественную речь человека. Она использует шаблоны в формате AIML (artificial intelligence markup language), который помогает

построить несколько разговорных правил. AIML часто используется для представления знаний.

```
<category>
  <pattern>WHAT IS YOUR NAME</pattern>
  <template><![CDATA[My name is <bot name="name"/>.]]></template>
</category>
<category>
  <pattern>WHAT ARE YOU CALLED</pattern>
  <template>
    <srai>what is your name</srai>
  </template>
</category>
```

Рис. 2. Фрагмент файла формата AIML

В настоящее время, специалисты попытались моделировать естественный язык разными методами, которые придумали на основе статистической математики. Перечислим несколько традиционных методов для построения модели естественного языка [9,11,13,21] таких как: N-Gramm, DF-IDF, HMM [9]. Эти модели можно применить в системах переводов [11,13,17,21] и в вопросно-ответных системах [17,21]. Самым современным алгоритмом моделирования языка является модель seq2seq [17,21] (рис. 3). Эта модель основана на рекуррентных сетях (RNN) [17,21] - рекуррентной сети кодирования и рекуррентной сети декодирования.

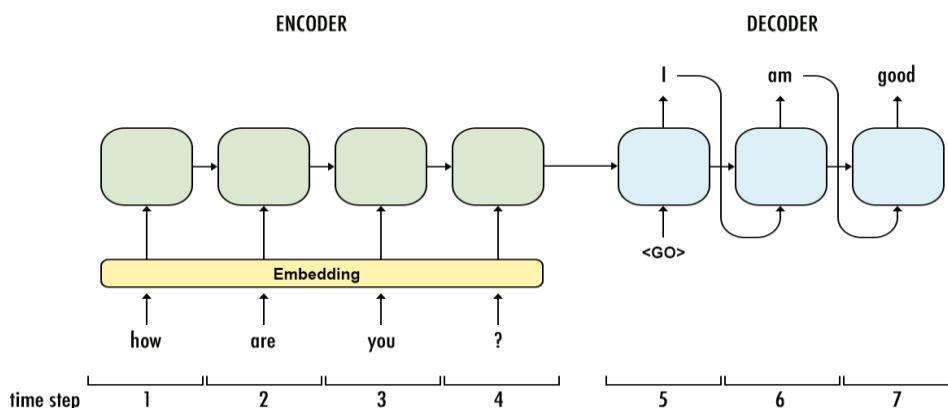


Рис. 3. Алгоритм “sequence to sequence” для создания модели естественного языка.

Существуют некоторые архитектуры для построения модели чат-бота. «Retrieval-based models» [29] использует эвристические шаблоны, в которых заранее записаны готовые ответы. Хотя в этой модели применяются некоторые алгоритмы машинного обучения для анализа запроса, но эта модель не генерирует новый ответ, а только выбирает из готовых ответов. Эта модель легко создается, ее архитектура изображена на рис. 4:

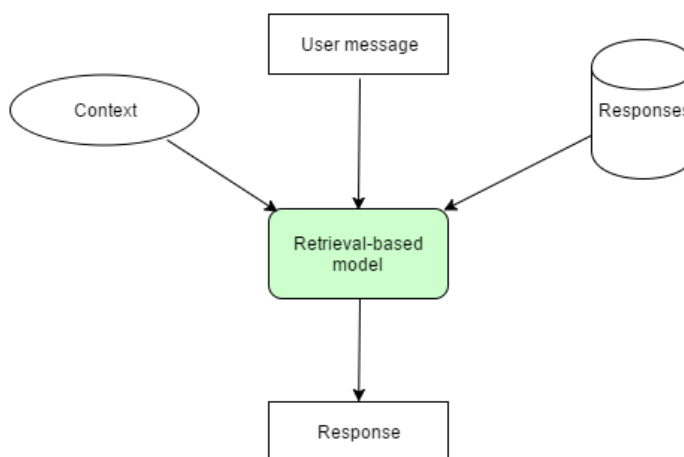


Рис. 4. Архитектура модели «Retrieval-based models».

Самая современная модель для построения модели чат-бота это «Генерирующая модель» (рис. 5), которая использует некоторые самообучаемые алгоритмы. Алгоритм «генерирующей модели» [17,21] основан на технологии машинного перевода. Однако эти модели трудно обучить, они часто делают грамматические ошибки (особенно в достаточно длинных предложениях) и, как правило, требуют огромных объемов обучающих выборок.

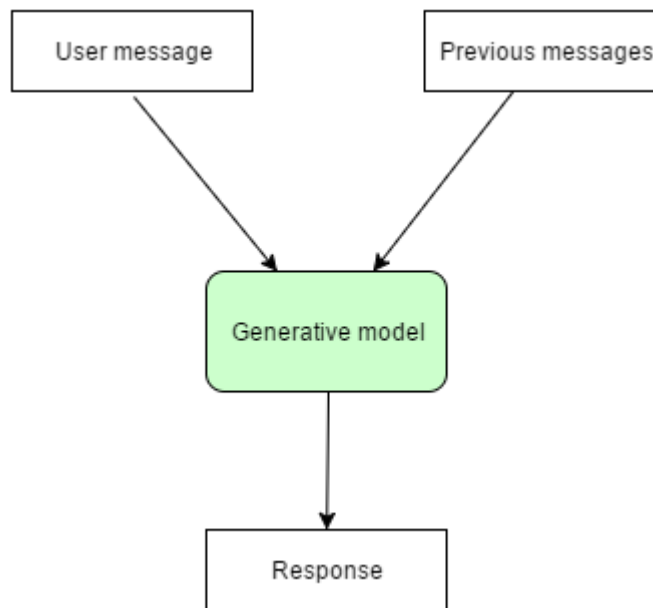


Рис. 5. Архитектура модели «Generative model».

На основе современных моделей естественного языка были созданы несколько умных чат-ботов. Самый известный чат-бот это “siri”, который создан компанией Apple. Программа появилась в 2010 и стала мощным катализатором для разработки чат-ботов с искусственным интеллектом. Через два года разработчики из Google создали «Google Now» для мобильного поисковика. В 2015 году появился «Alexa» от Amazon и в 2016 году - знаменитый Tay от Microsoft. Чат-бот Tay от Microsoft является самообучаемым ботом, в котором используется некий самообучаемый алгоритм. Он может обучаться естественной речи на ходу и пополнять свою базу знаний.

На вход этих систем необходимо подавать гигантские массивы языковых данных, которые анализируются системой и на основе которых моделируется язык по определенным принципам. Этот подход реализован, например, для машинного перевода в Googletranslate, за работой которой стоит очень мало работы лингвистов. Поэтому качество перевода этой системы хуже, чем системы традиционных методов. Чат-боты [17] – это программа, имитирующая человеческий разговор. Видно, что эволюция чаты-ботов сильно развивается и в

ИИ достигнуто много успехов. Но, по нашему мнению, эти работы относятся к ПИИ, которые выполняют задачи для хозяина - человека.

В 2014 году исследователями из Антверпенского университета создана роботизированная рука Aslan [30]. Система управления этой руки умеет переводить слова на язык жестов. Команда разработчиков этого проекта планирует сделать полноценного робота, который сможет обучаться языку жестов по данным, полученным с видеокамеры (глаза робота).

Среди программно-насыщенных систем всевозрастающую роль играют программные системы, осуществляющие управление различными объектами. При этом наиболее перспективными из них представляются адаптивные (самообучающиеся) системы, способные к самостоятельному приобретению необходимых для управления знаний на основе своего эмпирического опыта, на основе формального вывода новых знаний, а также на основе знаний, полученных от внешних источников через языковые интерфейсы. В частности, знания, которые программно-насыщенная система может приобрести в процессе общения с человеком, пополняя свою базу знаний, требуют, чтобы человеко-машинный интерфейс такой системы был в состоянии воспринимать обычную человеческую речь, извлекать из нее знания, ассоциируя воспринимаемую информацию с уже имеющимися элементами знаний, и, в соответствии с семантикой, записывать ее в структуру своей базы знаний. Такие программные системы (например, системы управления) могли бы, воспринимая от человека семантически достоверную информацию на его естественном языке, пополнять свою базу знаний, что постепенно повышало бы эффективность работы такой программы, например, качество управления (если это система управления). При этом речь идет не о системах, в которые человеком постоянно вводятся новые данные (например, экспертные системы), не о перепрограммируемых системах (например, на основе нечеткой логики), или системах, требующих обучения на обучающих выборках (например, нейросети), а о системах, способных обучаться, дообучаться или переобучаться непосредственно в процессе своего

функционирования, а именно – в процессе рабочего общения с человеком на его обычном языке. Для этого нужны специальные программные средства, обеспечивающие такое online восприятие новых для системы единиц естественной языковой информации, обучение их распознаванию, ассоциирование с соответствующими элементами уже имеющихся знаний, включение в структуру имеющихся в базе знаний сведений с последующим использованием этих автоматически приобретенных знаний.

Как показывает анализ публикаций, сегодня, практически, не существует программных средств, способных непосредственно в процессе работы осуществлять весь перечисленный выше список функций, необходимых для создания такого рода языкового человеко-машинного интерфейса. Основные «языковые» технологии сегодня связаны с использованием систем распознавания естественного языка, которые требуют предварительного обучения (например, искусственные нейросети). Но в своем рабочем режиме нейросети, как правило, уже не могут дообучаться и переобучаться. Поэтому возникает задача разработать принципы построения адаптивных программных систем с адаптивными языковыми интерфейсами, исследовать их возможности и оценить работоспособность. Настоящая работа посвящена именно таким проблемам.

Взяв за основу метод построения систем «автономного адаптивного управления» (ААУ), описанный в работах Жданова А.А., и предложенный там подход к созданию механизмов возникновения языковых эффектов в нейроподобном варианте реализации таких систем, автор данной диссертационной работы сосредоточил свои усилия на способах программной реализации этого подхода в виде языкового человеко-машинного интерфейса, на разработке соответствующих алгоритмов, структур данных, на устройстве базы знаний и т.д., которые позволили бы последователям реализовывать такого рода системы, развивать их и масштабировать. При этом все программирование проводилось в рамках нейросетевой парадигмы, с использованием самообучаемых моделей нейрона, разработанных Ждановым А.А. Термин «нейроподобный» используется в работе для того, чтобы подчеркнуть отличие

используемых моделей нейронов и сетей от известных «искусственных нейронных сетей», которые малопригодны для рассматриваемой задачи. Поскольку по результатам анализа литературы на эту тему, не было найдено подходящих приемов программирования, то основные решения оригинальны. В работе приводится также пример прототипа прикладной адаптивной системы (управление БПЛА) с адаптивным языковым интерфейсом.

Необходимо пояснить понятие «адаптивная система». Сегодня нет общепринятого трактования слова «адаптивный». Этим термином в технической литературе могут обозначать: а) наличие нескольких заранее предусмотренных вариантов режимов работы системы, предоставляемых на выбор пользователю, б) возможность предварительного обучения системы на обучающей выборке без последующего дообучения или переобучения, в) системы, настраиваемые разработчиками в соответствии с имеющимися условиями. В нашей работе мы под *адаптивностью* понимаем способность системы автоматически и в режиме online изменять свое функционирование в зависимости от текущих свойств системы или окружающей среды. Надо также заметить, что свойство адаптивности может реализовываться на нескольких уровнях системы – на уровне составляющих ее элементов (например, автоматическое изменение свойства нейронов), на уровне структуры программной системы (например, автоматическое изменение структуры сети нейронов), на уровне информации (например, на уровне информационного содержимого базы знаний) и на некоторых других уровнях. Адаптивность на любом из этих уровней делает систему адаптивной в целом. В настоящей работе использовались механизмы адаптации на всех трех указанных уровнях.

Поясним здесь также понятие «языка», имеющее тесную аналогию с лингвистическим определением языка. В теории метода ААУ показано, что в силу адаптивного характера системы ААУ, языковые функции ее интерфейса могут порождаться непосредственно в самой системе ААУ, что экспериментально и подтверждено в настоящей работе. Способность используемых адаптивных нейроноподобных элементов ассоциировать информацию из разных источников приводит к возможности ассоциирования в режиме online информации от

восприятия реальных объектов и ситуаций со звуковой, речевой, вербальной информацией, например, от человека, что и порождает *язык*. Под *языком* в данной работе подразумевается именно такая система речевых идентификаторов, которая соответствует образам реальных объектов в системе распознавания, образам действий в базе знаний и в подсистеме принятия решений, и образам качественных оценок. В лингвистике языковые идентификаторы образов объектов соответствуют существительным, языковые идентификаторы образов действий – глаголам, языковые идентификаторы качественных оценок – прилагательным. А в целом совокупность этих языковых идентификаторов соответствует известному понятию «второй сигнальной системы». Рассматриваемые языковые эффекты позволяют человеку при общении с такой программной системой постепенно выработать в ней ассоциации между словами человеческого языка и распознаваемыми системой объектами и ситуациями из окружающей среды. Выработанная в такой программной системе структура языковых ассоциаций, позволяет человеку влиять на принятие решений системой, вызывая с помощью языка распознавание в ней нужных человеку образов, действий или качественных оценок. Это дает человеку возможность подсказывать системе решения в затруднительных случаях и вызывать нужные действия в нужный человеку момент. При этом одновременно осуществляется персонализация такого интерфейса, поскольку учитываются именно характерные слова и произношение данного человека. Наличие такого человеко-машинного языкового интерфейса способно повысить эффективность работы систем, за счет всех тех аспектов, которые может дать язык общения – подсказки, обучение, передача знаний, формирование нужных качественных оценок и т.п. Как сказано выше, принципы построения такого рода актуальных языковых систем описаны в работах Жданова А.А. Возникает проблема и соответствующие ей задачи реализации этих принципов в программных решениях: поиски подходящей архитектуры программной системы, разработка алгоритмов, структур данных, кодирование и некоторые другие задачи, решению которых и посвящена настоящая работа.

Чтобы построить систему управления с языком, требуется построить систему распознавания знаков (единиц языка), в частности, это требует создания системы распознавания звуков. Существует много методов для построения систем распознавания звуков, особенно с применением глубокого обучения [16] с высокой точностью распознавания. Но опять, для обучения этих алгоритмов требуются гигантские обучающие выборки языковых данных, а после обучения невозможно добавить новые образы. Сегодня отсутствуют алгоритмы распознавания образов, у которых процесс обучения мог бы реализовываться с малыми данными, обучение было бы быстрым, и они могли бы переобучаться при добавлении новых образов.

Коротко опишем систему ААУ [1,2,5], которая явилась базовой технологией в настоящей работе. В системе ААУ осуществляется попытка построения модели мозга, отражающей его функциональный состав и структуру, разработан алгоритм его работы, который многократно реализовывался и показал свою работоспособность. Схема системы ААУ показана на рис. 6.

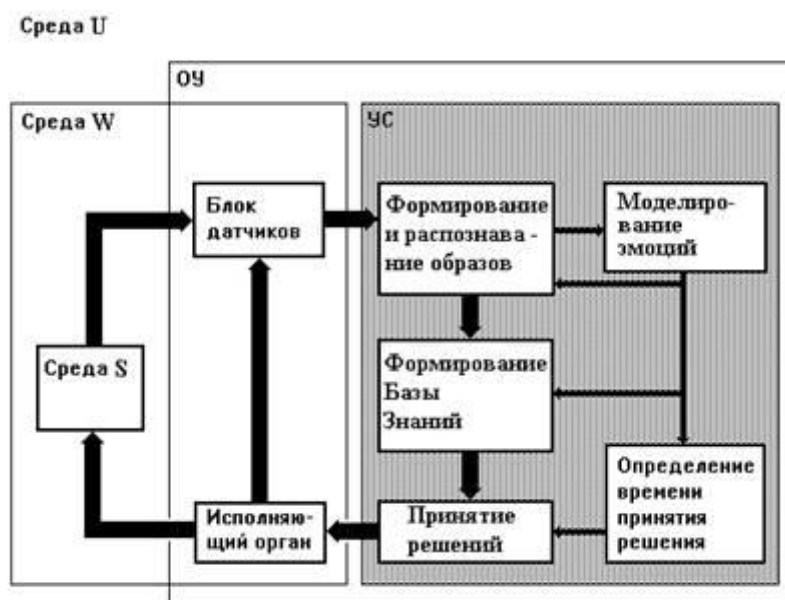


Рис. 6. Общая схема системы ААУ.

Система ААУ является объединенным адаптивным распознающее-обучаемым комплексом, в составе которого непосредственно в процессе

управления в двух параллельно работающих контурах решаются задачи обучения и управления, а именно - формирование новых образов, поиск и формирование новых знаний, нахождение качественных оценок для образов, распознавание образов, принятие решений и некоторые другие задачи.

Система ААУ является адаптивной системой, поскольку все подсистемы этой системы тоже являются адаптивными. При этом адаптивность может реализовываться на разных уровнях этой системы – на уровне ее элементов (например, нейронов), на уровне структуры программы (например, структуры сети нейронов), на информационном уровне (например, на уровне содержимого базы знаний). Например, блок формирования и распознавания образов (ФРО) метода ААУ состоит из датчиков, нейронов и связей между ними, образующими нейроноподобную сеть подсистемы ФРО. Подсистема ФРО решает задачу формирования новых образов (задача автоматической классификации без учителя), процесс обучения ФРО является процессом генераций структуры нейроноподобной сети и обучения ее нейронов. За счет формирования новых образов, «база знаний» системы ААУ тоже начинает накапливать новые знания и сохраняет их в своей памяти. Это значит, что система управления методом ААУ способна к адаптации к изменениям среды, и мы заранее не знаем точно структуры этой системы, потому что она является самоорганизующейся системой, структура которой во многом зависит от ее конкретного опыта. Поэтому мы не можем прямо передать опыт (знания) из одной «живой системы» управления в другую систему управления (у получателя знаний может оказаться иная структура нервной сети), а только косвенно через среду с помощью языка (рис. 7.).

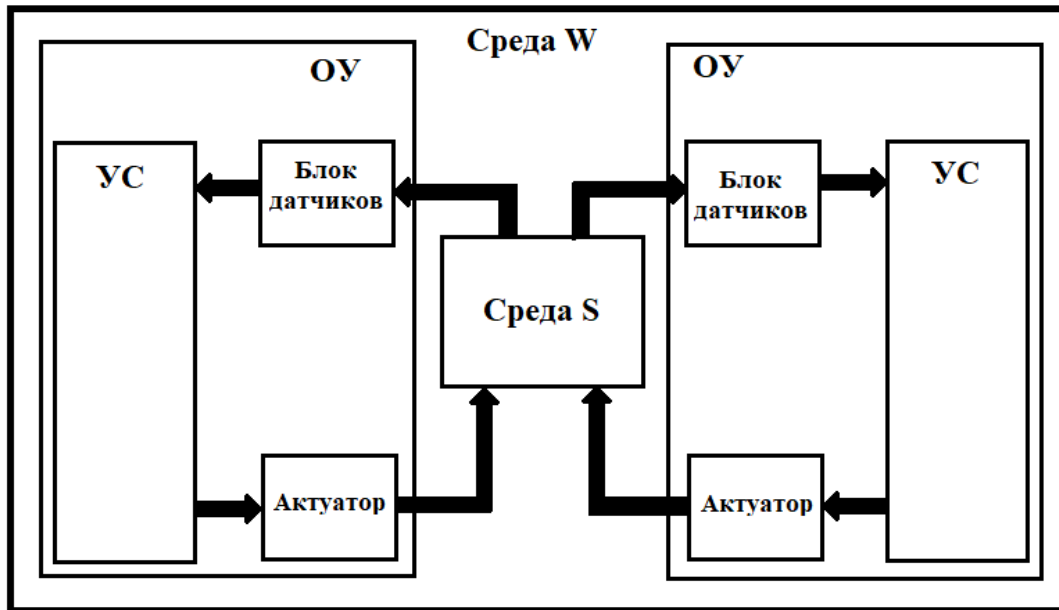


Рис. 7. Схема передачи знаний с помощью языка в системе «среда – ОУ – ОС».

Описание модели нейронных конструкций в методе ААУ, порождающих языковые явления

Если смотреть на среду как на множество элементов, которые связаны в некоторую структуру, то объект управления (ОУ) – это подмножество элементов среды. По условию автономности [1,2,5], управляющая система (УС) находится в ОУ и управляет, согласно целевым функциям. “Главными целевыми функциями нервной системы всякого организма являются: выживание организма и накопление знаний его нервной системой” [5]. Тогда искусственные управляющие системы, построенные по методике ААУ, также имеют целевые функции:

- выживание ОУ;
- накопление знаний о системе «среда – ОУ – УС».

Систему «среда – ОУ – УС» можно представлять в виде множественных отношений:

$$УС \subset ОУ \subset среда,$$

или в виде кругов Эйлера:

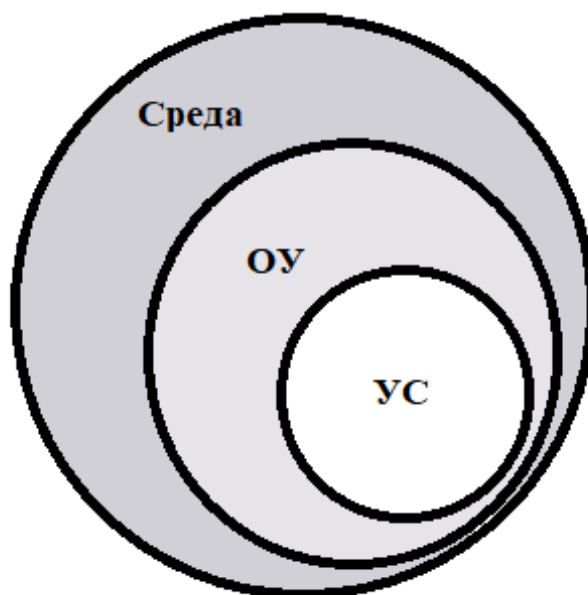


Рис. 8. Представление системы с автономным объектом управления в виде кругов Эйлера [5].

Можно сказать, что любое взаимодействие со средой, которое происходит в системе «среда – ОУ – УС» можно интерпретировать как следующую импликацию:

(условие & действие) → следствие → эмоциональная оценка следствия.

Эмоциональную оценку «следствия» целесообразнее относить к функциям аппарата эмоций организма, отделив их от перехода от состояния к состоянию – от условий к следствиям. Элементы, из которых состоит среда, ОУ и УС, закономерно переводят свои входные воздействия в выходные (в рассматриваемом понятийном срезе). Если элемент только поглощает, но ничего не возвращает, то он называется «стоком». Если элемент генерирует воздействия, не являющиеся закономерным ответом на входные воздействия, то он называется «истоком» [5]. Стоки и истоки есть в среде, ОУ и УС, в обобщенном виде они показаны на рис. 5.

Если УС обнаруживает, что воздействия вида *(условие & действие) → следствие* образуют обратную связь и закономерны, то УС сохранит

информацию о них в своей памяти - *базе знаний*. Очевидно, что «истоки» и «стоки» не только бывают в среде, но и в теле ОУ, и в самой в УС. Все маршруты воздействий, возможных в управляемой системе, изображены на рис. 9.

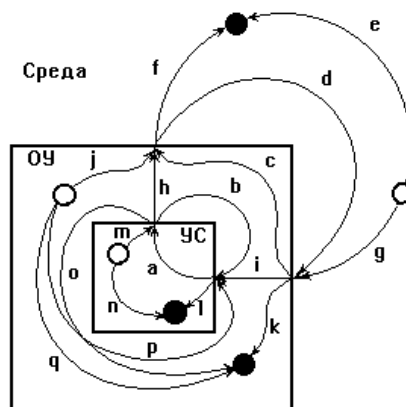


Рис. 9. Концептуальная схема возможных воздействий в управляемых системах [5].

Все маршруты подробно описаны в [5]. Основной цикл управляемого взаимодействия это « $-d-i-a-h-d-$ », в нем УС должна найти закономерные последовательности (*условие & действие*) \rightarrow *следствие*, с помощью которых можно управлять в системе «среда – ОУ – УС».

Языковая система тоже является подмножеством в среде со своей структурой. Очевидно, что прообразы, наблюдаемые УС, могут быть языковыми и эти прообразы тоже участвуют в закономерных последовательностях (*условие & действие*) \rightarrow *следствие* по циклу « $-d-i-a-h-d-$ ». УС также может находить эти закономерные последовательности и записывать в свою память. Если УС нашла пару образов, которые часто одновременно поступают в УС, то в УС может быть сформирован новый образ, распознающийся, как только один из них будет распознан. Тогда один из этих образов заменяет, идентифицирует собой другой образ, и порожденный образ является ассоциативным образом. Множество ассоциативных образов мы можем разделить на два подмножества. Если одно из

них может легко воспроизводиться ОУ, то это подмножество может являться множеством идентификаторов, и составить основу «языка». По нашему мнению, с одной стороны, язык повышает качество управления одной УС, так как он позволяет перейти к многошаговому принятию решений, а с другой стороны, язык является средством передачи информации для управления в коллективе объектов.

Синтез самообучаемой подсистемы распознавания звуков в системе ААУ

При создании УС, работающей с языком, требуется построение подсистемы ФРО, способной формировать и распознавать языковые образы. В настоящее время, направление систем распознавания речи сильно развивается. В общем, каждая задача распознавания речи сводится к тому, чтобы выделить, классифицировать и соответствующим образом отреагировать на человеческую речь из входного звукового потока. Существуют разные методы распознавания речи, такие как метод статистики и нейронные сети с применением глубокого обучения [14]. Но точность распознавания у этих систем не высокая и требуются большие данные для обучения.

В данной работе речь пойдет о моделировании начала языкового тракта, а именно – о процессе распознавания звуковой речи с помощью биоподобных рецепторов и биологически инспирированной системы распознавания отдельных слов на базе нейронов, разработанных в теории ААУ.

Известно, что нервные системы живых организмов построены, как можно видеть, на универсальной для всего живого элементной базе – нейронах и сетях из таких нейронов. В данной работе мы представляем новый подход к распознаванию речи на основе нейроноподобных сетей А.А. Жданова [1,5], существенно отличающихся от «искусственных нейронных сетей». Мы предприняли попытку построения системы распознавания речи с помощью только конструкций из нейронов, не используя, по возможности, решений в форме программ фон-неймановского типа.

Цель и задачи исследования

Целью диссертационной работы является разработка и исследование способов реализации нового типа программных систем, содержащих в себе средства автоматического адаптивного порождения языкового человеко-машинного интерфейса непосредственно в подсистеме распознавания, на основе использования метода автономного адаптивного управления и соответствующих ему адаптивных нейроноподобных элементов. Достижение поставленной цели потребовало решения следующих задач:

1. разработать и исследовать алгоритмы и структуры данных для построения самообучаемых систем распознавания звуковых образов на основе адаптивных нейроноподобных элементов и сетей из метода ААУ;
2. разработать и исследовать алгоритмы построения распознающих нейроноподобных сетей, способных к ассоциированию звуковых образов с образами реальных объектов и ситуаций;
3. разработать и исследовать алгоритмы автоматического выращивания нейроноподобных сетей для самообучаемой системы распознавания (подсистемы формирования и распознавания образов (ФРО)) для формирования распознавания образов статических и динамических объектов;
4. разработать и исследовать новую конструкцию базы знаний системы ААУ на основе адаптивных нейроноподобных элементов для сохранения семантически достоверных знаний, которая обеспечивает совместность работы с конструкцией, порождающей язык;
5. реализовать адаптивный языковой человеко-машинный интерфейс для программной системы управления ААУ на примере прикладной системы автономного адаптивного управления для БПЛА.

Метод исследования

В данной работе использованы методы теории вероятности, математической статистики, теории множеств, методы вычислительной математики, метод автономного адаптивного управления.

Научная новизна

Новизна данной диссертационной работы состоит в следующем:

1. Впервые предложен, разработан и запрограммирован новый метод и алгоритмы построения языкового человеко-машинного интерфейса на основе подходов и адаптивных нейроноподобных элементов, разработанных Ждановым А.А. Разработаны новые алгоритмы построения адаптивной системы распознавания произносимых слов на основе адаптивных нейроноподобных элементов, специальных сетевых структур фильтров с временными задержками.
2. Впервые разработаны нейроноподобные сетевые программные конструкции для ассоциирования звуковых языковых идентификаторов с образами реальных объектов в базе знаний управляющей системы.
3. Впервые предложены и применены алгоритмы автоматического роста нейроноподобных сетей из адаптивных нейроноподобных элементов в самообучаемых системах распознавания образов (подсистемы ФРО в системах ААУ), основанные на «деревьях событий».
4. На примере разработанной прикладной автономной адаптивной системы управления БПЛА (коптерами), реализующей разработанные принципы построения адаптивного языкового интерфейса, показаны:
а) возможности эффективного применения алгоритма построения нейроноподобной сети языкового интерфейса, приводящего к повышению качества управления и б) показан пример адаптивного управления объектом звуковыми командами через адаптивный языковой интерфейс.

Теоретическая ценность

Разработаны алгоритмы построения адаптивного нейроподобного языкового человеко-машинного интерфейса для программных систем, способных автоматически обучаться распознаванию часто произносимых человеком на устном языке слов, и ассоциировать их с одновременно распознаваемыми образами реальных объектов или ситуаций. Программные системы такого типа могут широко использоваться при построении программного обеспечения для робототехнических комплексов, для программно-насыщенных вещей и для других программных систем, требующих языкового человеко-машинного интерфейса. Система языковых идентификаторов, распознаваемая таким интерфейсом, при достаточно большой мощности словарных множеств будет представлять собой язык общения человека и машины, который может быть использован человеком для управления данной технической и программной системой. Важными компонентами проведенной разработки являются также адаптивные алгоритмы системы распознавания произносимых слов, в том числе алгоритмы автоматического выращивания нейроподобных сетей.

Практическая ценность

Практическая ценность полученных результатов состоит в том, что разработанные адаптивные алгоритмы можно применять при разработке различных адаптивных программных систем и их компонент. В работе описан также (глава 4) прикладной пример практического применения разработанного языкового человеко-машинного интерфейса для управления БПЛА, демонстрирующий заявленные эффекты языкового интерфейса.

Основные положения, выносимые на защиту:

1. Алгоритм построения системы распознавания статических и динамических образов на основе нейроподобных сетей, в частности - система

распознавания звуков для построения языкового человеко-машинного интерфейса.

2. Алгоритм «выращивания» нейроподобных сетей для построения самообучаемой системы формирования и распознавания образов в системе ААУ на основе дерева событий.
3. Метод построения базы знаний на основе нейроподобных сетей из нейронов, описанных в методе ААУ.
4. Алгоритм автоматического формирования ассоциативной памяти на основе нейроподобных сетей для построения адаптивного языкового человеко-машинного интерфейса.
5. Программная система автономного адаптивного управления БПЛА с применением языкового человеко-машинного интерфейса.

Апробация работы и публикации

По материалам диссертации опубликовано 6 работ, из них: 3 – в российских журналах, входящих в перечень ВАК, 2 в трудах конференций и 1 свидетельство о регистрации программ на ЭВМ.

Основные положения работы докладывались на следующих конференциях:

1. 2-я международная конференция «Инжиниринг & Телекоммуникации» - En&T 2015;
2. 4-я международная конференция «Инжиниринг & Телекоммуникации» - En&T 2017.

Основные положения работы опубликованы в следующих журналах и трудах конференций:

1. Нгуен Нгок Зиеп, Мань Ха Ле. "Нейросетевой метод снятия омонимии." Труды Московского физико-технического института 7.4 (2015): 174-182.

2. Чан Ван Нгиа, Ву Вьет Тханг, Нгуен Нгок Зиеп, реализация нейронной сети на fpga для распознавания атаки на информационные системы, II Международная конференция «Инжиниринг & телекоммуникации En&T 2015».

3. Степанян И.В., Зиеп Н.Н. Растущие свёрточные нейроподобные структуры для задач распознавания статических образов // Нейрокомпьютеры: разработка, применение. 2018. № 5. С. 5-14.

4. Зиеп, Н.Н., Жданов, А.А. нейроподобный подход к распознаванию речи. Программирование, 2018 (3), pp.49-62.

5. Нгуен Нгок Зиеп, Мишустин И.А. Принцип системы распознавания образов на основе нейроподобных сетей. IV Международная конференция «Инжиниринг & телекоммуникации En&T 2017» С. 53-54.

6. Жданов А.А., Нгуен Н.З., Перский Г.С., Пешенко Р.Э., Пижонков А.Г., Степанян И.В., Сямиуллин З.С., "программа динамического синтеза растущих биоподобных структур на основе искусственных нейронов с расширенной функциональностью и способностью к дообучению для задач распознавания образов и управления динамическими объектами". Свидетельство о регистрации программ на ЭВМ РФ № 2018661458.

В 1-й публикации автором предложен метод представления естественного языка для снятия омонимии русского языка. Во 2-й публикации представлена разработанная автором реализация нейронной сети на FPGA. В 3-й и 5-й публикациях автором описана нейроподобная структура фильтров для построения самообучаемой системы распознавания статических образов на основе нейроподобных сетей. В 4-й публикации описан разработанный автором метод построения системы распознавания звуков на основе нейроподобных сетей. В 6-й публикации описана разработанная автором программа реализации алгоритма роста нейроподобных сетей на основе «дерева событий».

Личный вклад автора

Все представленные в диссертации результаты получены лично автором.

Структура и объем работы диссертации

Диссертационная работа состоит из введения, четырех глав, заключения, списка использованной литературы. Работа содержит 130 страниц основного текста, 3 таблицы, 89 рисунок. Список использованной литературы включает 51 наименований.

ГЛАВА 1. Описание адаптивных нейроноподобных программных элементов и конструкций, порождающих языковой человеко-машинный интерфейс в программной системе на примере системы автономного адаптивного управления

1.1. Описание элементов подсистемы ФРО метода ААУ

В системе ААУ подсистема *формирования и распознавания образов* (ФРО) обеспечивает распознавание более сложных образов, чем то, что могут дать датчики-рецепторы (которые, как правило, измеряют только текущее значение величины $x(t)$ или ее производную $dx(t)/dt$). Подсистема ФРО состоит из выходов рецепторов и нейронов, которые соединены межнейронными связями через синапсы. Качество управления системы ААУ сильно зависит от качества формирования и распознавания образов подсистемой ФРО. Из условия автономности [5] следует, что нейроноподобные сети подсистемы ФРО должны решать задачи самообучаемой системы распознавания:

- формирование образов;
- распознавание сформированных образов в реальном времени.

Классическая искусственная нейронная сеть (ИНС) обычно используется как обучаемая с учителем система распознавания и классификации [44]. «Формальный нейрон» в ИНС (так называемый нейрон Маккалока-Питтса) – это вычислительная единица, которая получает входной сигнал и производит над ним вычисления – вычисляет взвешенную сумму входных сигналов и сравнение ее с порогом, и передает его дальше. ИНС, основанные на таких «формальных нейронах», описываются следующим способом (рис. 1.1).

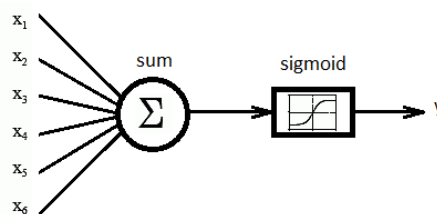


Рис. 1.1. Структура классического нейрона из ИНС.

Часто используемая функция активации – сигмоида:

$$f(z_i) = \sigma(z_i) = (1 + e^{-z_i})^{-1}. \quad (1.1)$$

Рассмотрим регулярную структуру традиционной ИНС (рис. 1.2) на основе «формальных нейронов». В такой сети каждый нейрон рассматриваемого слоя соединен со всеми нейронами предыдущего слоя.

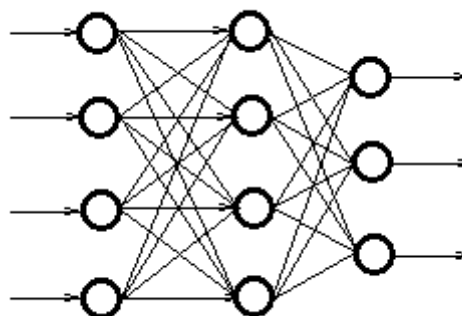


Рис. 1.2. Структура традиционной искусственной нейронной сети.

Обучение традиционной ИНС производится некоторым способом, например, методом обратного распространения ошибки (back propagation). В качестве такой функции ошибки может быть использована сумма квадратов отклонений выходных сигналов сети от требуемых [44]:

$$C = \sum_i^n (y_i - y_i')^2 \quad (1.2)$$

Метод распространения ошибки состоит из двух этапов:

- вычисление ошибки;
- коррекция весов нейронов.

Заметим, что при этом необходимо знать «требуемые» в текущей ситуации выходные сигналы, и неясно, кто в реальной жизни может знать это! Отсюда следует, что процесс обучения ИНС является коррекцией весов нейронов сети, структура которой фиксирована. При этом возникает большая проблема, состоящая в том, что нет возможности добавления в обученную ИНС новых классов (образов). При добавлении новых классов, надо переобучить значения весов всех нейронов и, при необходимости, перестроить структуру сети – добавить нейроны в слои.

Теперь рассмотрим биологическую нейронную сеть и структуру биологического нейрона (рис. 1.3.). Обычно, биологический нейрон имеет несколько (иногда до ~1000) входов и только один выход. Из биологических сведений известно, что отдельный нейрон не соединен со всеми, но только с несколькими рецепторами или выходами других нейронов, в отличие от классических ИНС. Поэтому очевидно, что традиционные ИНС явно не соответствуют биологическим нервным сетям.

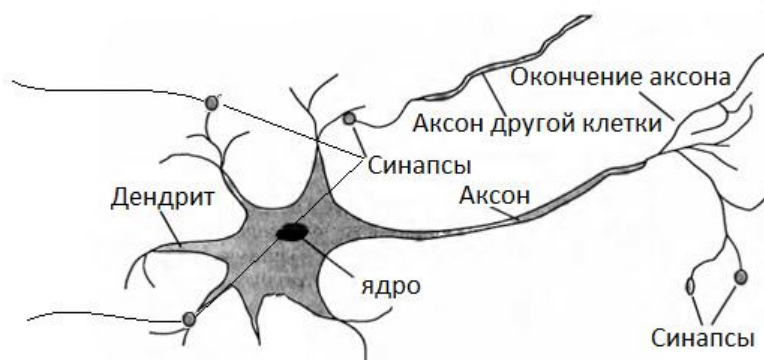


Рис. 1.3. Структура биологического нейрона.

Биологический нейрон (рис. 1.3) является основным блоком нервной системы. Он является клеткой, подобной всем другим клеткам тела; однако определенные существенные отличия позволяют ему выполнять вычислительные функции и функции связи внутри мозга. Нейрон состоит из трех основных частей: тела клетки, дендритов и аксона, каждая часть со своими взаимосвязанными

функциями. На входных отростках нейрона – дендритах расположены синапсы, через которые нейрон получает сигналы от других клеток – нейронов или рецепторов. Существуют два типа рецепторов – «тонические» и «фазические» рецепторы [5]. Через тонические рецепторы передается информация о текущем значении наблюдаемого воздействия (0-й производной), а через фазические рецепторы передается информация об изменении воздействия (1-й производной). Из рецепторов приходят сигналы (возбуждение) в тело клетки и там «вычисляется» суммарный потенциал (суммарное возбуждение) тела клетки. Согласно общепринятому пониманию, при котором нейрон понимается просто как пороговый сумматор, если суммарное значение возбуждения превышает пороговое значение (приблизительно -50 мВ), то клетка возбуждается и изменяется свою полярность. Через некоторое время, потенциал достигает потенциала действия около $+30$ мВ и в аксонном холмике вырабатывается импульс, который по аксону передается на следующие клетки. Однако мы полагаем, что нейрон не есть просто пороговый сумматор, а представляет собой самостоятельную самообучаемую систему распознавания образов, и в описанный режим порогового суммирования он переходит только на некоторой фазе своей жизни – после завершения процесса самообучения. Процесс возбуждения нейрона показан на рисунке 1.4:

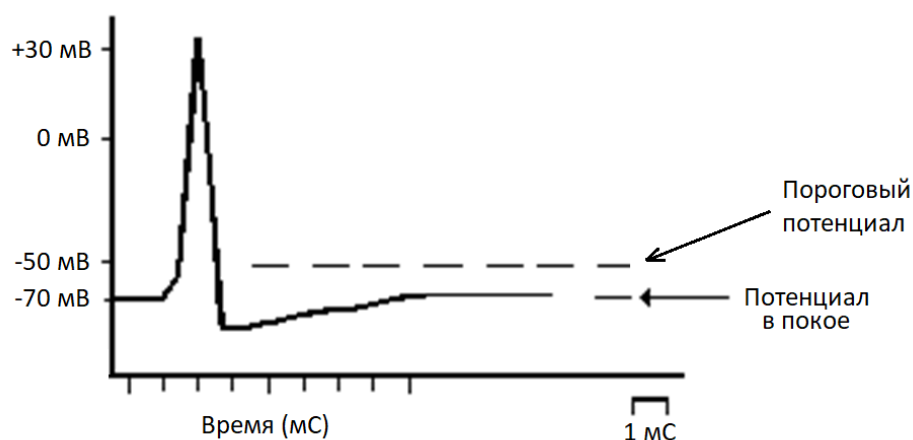


Рис. 1.4. Нервный импульс.

В биологии говорят, что нервный импульс либо есть, либо его нет, это означает, что с точки зрения кибернетики мы можем кодировать наличие и отсутствие нервных импульсов сигналами «1» и «0». То есть, нервные импульсы подчинены закону «все или ничего». Тогда нервные импульсы можно представлять таким образом:

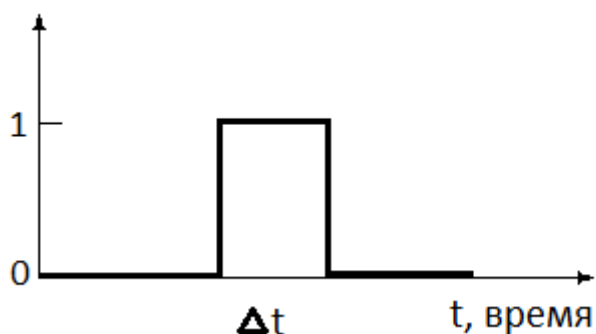


Рис. 1.5. Модель импульсного сигнала.

Напомним, что биологический нейрон соединен с выходами несколько рецепторов или других нейронов. Это значит, что сама конфигурация входных связей нейрона уже определяет «пространственные» характеристики тех прообразов, которые может анализировать данный нейрон. Кроме того, если нервные импульсы поступают на входы этого нейрона, пройдя через некоторые нейроны, при этом каждый нейрон имеет временную задержку, то структура нервных импульсов несет еще «временные» характеристики, которых у формальных нейронов из ИНС нет. Временные задержки в нейронах обусловлены синаптическими задержками, конечной скоростью распространения нервных импульсов по нервным волокнам разной длины и другими причинами. В этом отношении типовая модель ИНС, в которой временные задержки непосредственно не учитываются, тоже мало соответствует своему биологическому прообразу.

Нейроноподобная сеть из метода ААУ. Отличающиеся от формальных нейронов ИНС, новые модели нейронов [3] (рис. 1.6.) были представлены

профессором Ждановым А.А. в описании теории автономного адаптивного управления (ААУ). Они в большей степени обладают характеристиками биологических нейронов, чем «формальные нейроны» из ИНС. В соответствии с функциями биологических нейронов, нейроны из теории ААУ умеют решать две задачи:

- находить некоторые закономерные явления на входах и формировать образы;
- распознавать сформированные образы при поступлении информации от их прообразов на входы.

Помимо этого, в данных моделях нейрона учитываются синаптические временные задержки.

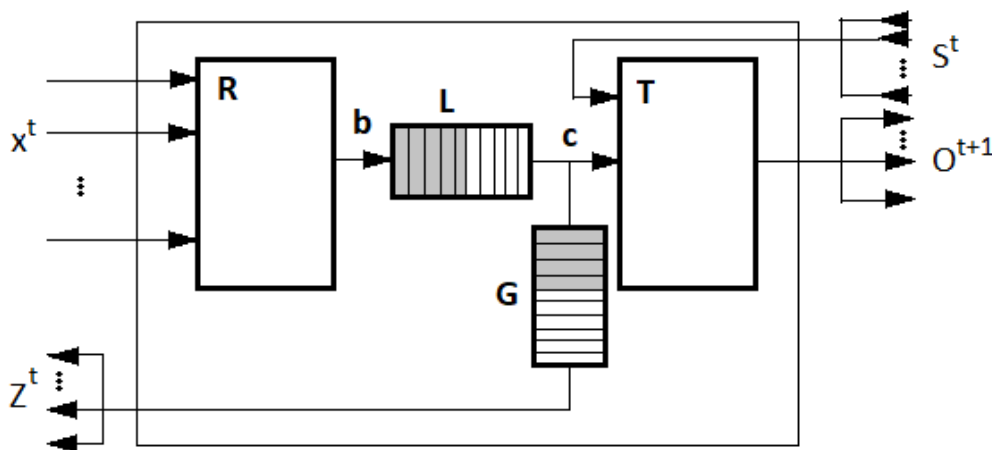


Рис. 1.6. Модель самообучаемого нейрона Жданова типа 1.

На рисунке 1.6:

- $x^t = (x_1, x_2, \dots, x_n)^t$ – Входной вектор в момент t , значение элементов которого принадлежат множеству $\{0,1\}$, это значит, что все синапсы нейрона могут принимать только значение 0 или 1.
- «Структурное правило» - Элемент R обеспечивает восприятие входных импульсов. В нем выполняется операция проверки «пространственно-временных» характеристик входной информации.

Пример, самое простое «структурное» правило R – логическое «И», которое проверит наличие одновременно всех входных импульсов. Событие $b = 1$ когда сигналы удовлетворяют структурному правилу элементу R .

$$b = R(x_1, x_2, \dots, x_n)$$

- «Статистическая гипотеза» - Элемент L осуществляет подсчет N событий « $b = 1$ », чтобы проверить, случайно или неслучайно появление события « $b = 1$ ». Если число $N < M$ (пороговое значение), элемент L не пропускает сигнал в точку «с». В случае $N \geq M$, элемент L пропускает сигнал и можем говорить, что данный нейрон становится обученным, так как данный прообраз прошел статистическую проверку.

$$l = \begin{cases} 0, & \text{if } N < M \\ 1, & \text{if } N \geq M \end{cases} \quad (1.3)$$

- «Триггер» - Элемент T включает сигнал «с» и удерживает этот сигнал на выходе $O = 1$. Элемент T работает как кратковременная память, которая говорит о том, что образ распознан, и нейрон помнит его некоторое время.
- «Тормозящий вход» - специальный вход, по которому поступает отключающий сигнал “ S ”. Если появился отключающий сигнал « $S=1$ », то «триггер» T выключит на следующем такте выходной сигнал $O = 0$.

Работу нейрона можно описать следующим логическим выражением:

$$O^{t+1} = \neg S^t \& ((b^t \& l^t) \vee O^t) \quad (1.4)$$

Процесс обучения нейрона

Сначала нейрон является необученным. Если нейрон обнаружит входной вектор, который удовлетворяет «структурному правилу» R , то возникает сигнал b

= 1. Сигнал $b = 1$ поступает на проверку в блок «Статистическая гипотеза» L , который проверяет закономерность события $b=1$. Если число N события « $b=1$ » превзойдет порог M , заданный в элемент L , то нейрон становится обученным.

1.2. Описание структуры нейроноподобных фильтров

При наблюдении реальных образов можно формировать большое количество образов по разным признакам - по цвету, форме и т.д. Если используется самообучаемая система распознавания (автоматический классификатор), то следует указывать правила формирования образов (правила классификации), потому что это зависит от желания заказчика такой системы. Такие правила могут задаваться в различном виде. В нейросетевых структурах роль этих правил выполняет сам набор и состав датчиков, нейронов и структура сети. Например, можно использовать опорные структуры. Опорные структуры помогают получить более понятные признаки образов. Они играют роль фильтров, и в результате фильтрации мы получим эффективные признаки, используемые для группирования образов или построении слоев признаков. Построение карт признаков осуществляется разными элементами нейроноподобной сети, такими как рецепторы и нейросетевые фильтры.

Например, из двух типов рецепторов «тонического» (рис. 1.7а) и «фазического» (рис. 1.7б) мы можем получить две карты признаков образа, иллюстрации этих импульсов изображены следующим способом (рис. 1.7).

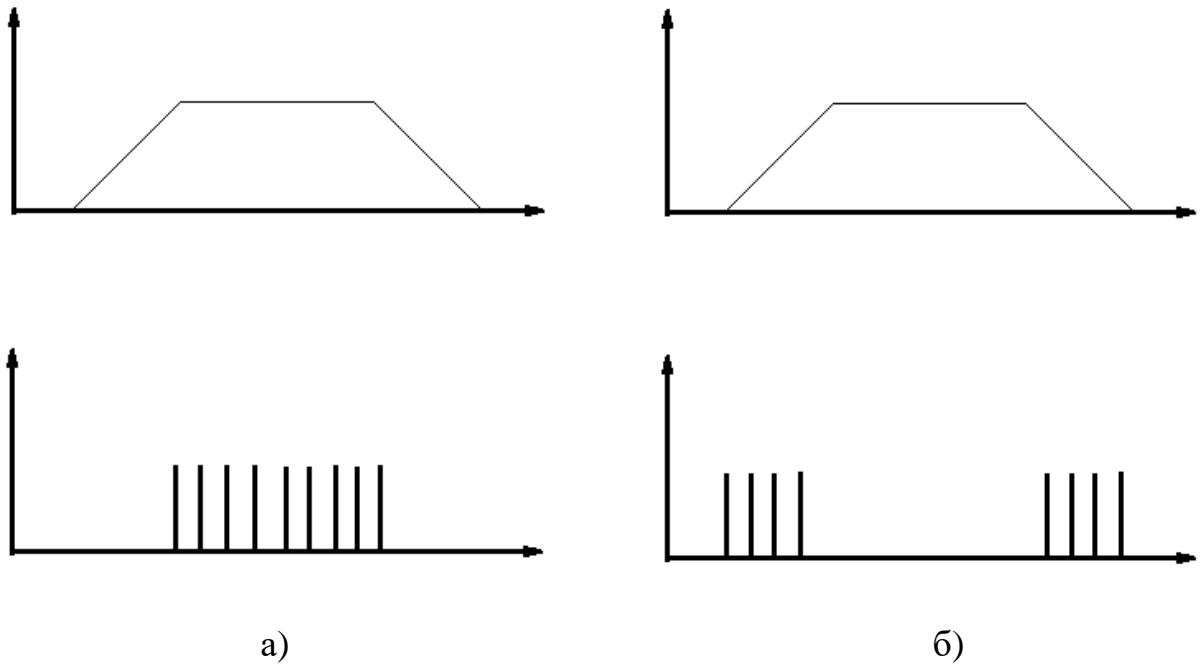


Рис. 1.7. а) тонический и б) фазический датчики.

В данной работе мы хотели симитировать структуры зрительной коры мозга [31] для построения подсистемы ФРО на основе нейроноподобных сетей ААУ. Чтобы применить опорные структуры, необходимо создать нейросетевые фильтры, через которые будут строиться карты признаков для распознавания.

Нейроноподобный фильтр. Нейроноподобный фильтр служит для фильтрации набора значимых сигналов и исключения набора незначимых [46]. Эта структура должна работать как следующий логический оператор (рис. 1.8.).

$$x_1 \wedge (\neg x_2) \approx$$

$x_1 \backslash x_2$	0	1
0	0	0
1	1	0

Рис. 1.8. Таблица выполнения логического оператора [46].

Здесь x_1 - набор значимых сигналов, x_2 - набор незначимых сигналов. С помощью данного фильтра мы на выход фильтра пропускаем только набор (1,0), а

все остальные наборы (0,0) (0,1) (1,1) фильтруются. Этот логический оператор реализован в виде структуры, показанной на рис. 1.9. Отметим, что применение тормозной связи достаточно для реализации этого же функционала.

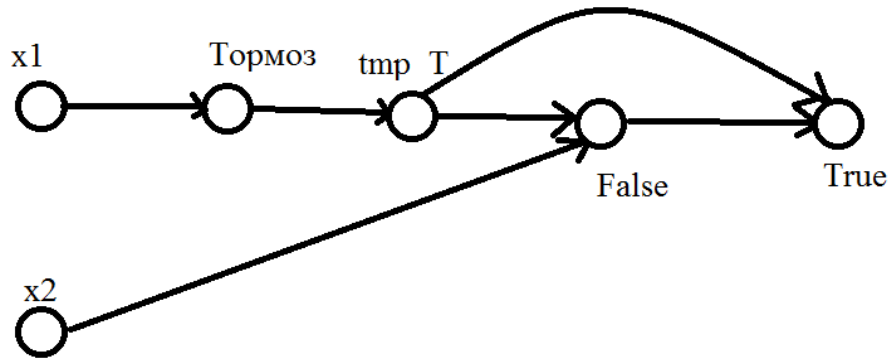


Рис. 1.9. Структура фильтра, построенного на основе нейронов типа 1 [46].

Для анализа работы данного фильтра в соответствии с приведенной формулой рассмотрено его функционирование для каждого случая. Возможно 3 случая, которые приведены на рис. 1.10.

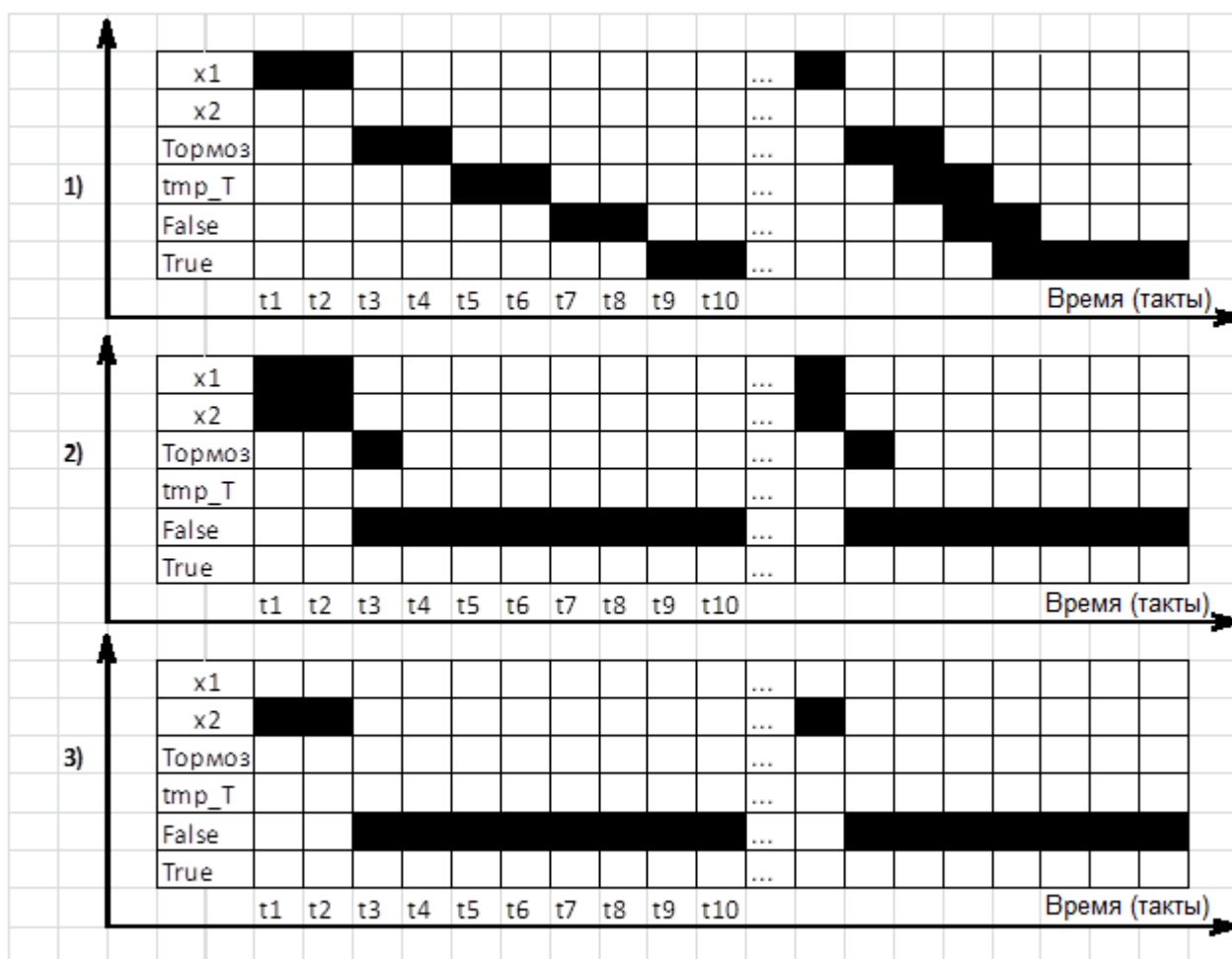


Рис. 1.10. Временные диаграммы, отражающие реализацию формулы для каждого из 3-х случаев. По оси ординат отложены идентификаторы нейронов, по оси абсцисс – время в тактах. Зачерненные ячейки соответствуют активности нейронов по времени [46].

Нейросетевая структура очереди. По нашему убеждению, динамический образ - это пространственно-временные комбинации сигналов. Задача распознавания динамических образов состоит из двух задач:

- распознавания векторов всех элементов пространственно-временного образа;
- распознавания последовательности распознанных элементов.

Существуют два метода для решения второй задачи:

- использовать нейрон 2-го типа; (рис. 1.11)
- использовать структуру очереди на основе нейрона 1-го типа.

Нейрон 2-го типа [5]. Как сказано выше, в нейроне 1-го типа [5] временные задержки учтены только в самом общем виде в том, что нейрон срабатывает на следующем такте после получения входного вектора, значит, он имеет кратковременную память. К тому же, сигнал на выходе триггера тоже висит долго, пока его не отключат. Обращаясь к биологическому нейрону, мы видим, что временные задержки и временные характеристики импульсов играют важную роль в работе биологических нейронов. Поэтому чтобы более детально учесть основные временные задержки в работе биологического нейрона, создана модель нейрона 2-го типа (рис. 1.11) [5]. Модель нейрона 2-го типа получается декомпозицией нейронов 1-го типа, подробнее описание модели нейрона 2-го дано в [5]. В этой модели нейрона, сигналы поступают на входы нейрона, а потом через блок задержки и блок генерации импульса идут на блок проверки R. Если нейрон сработал, то пространственно-временный образ распознан.

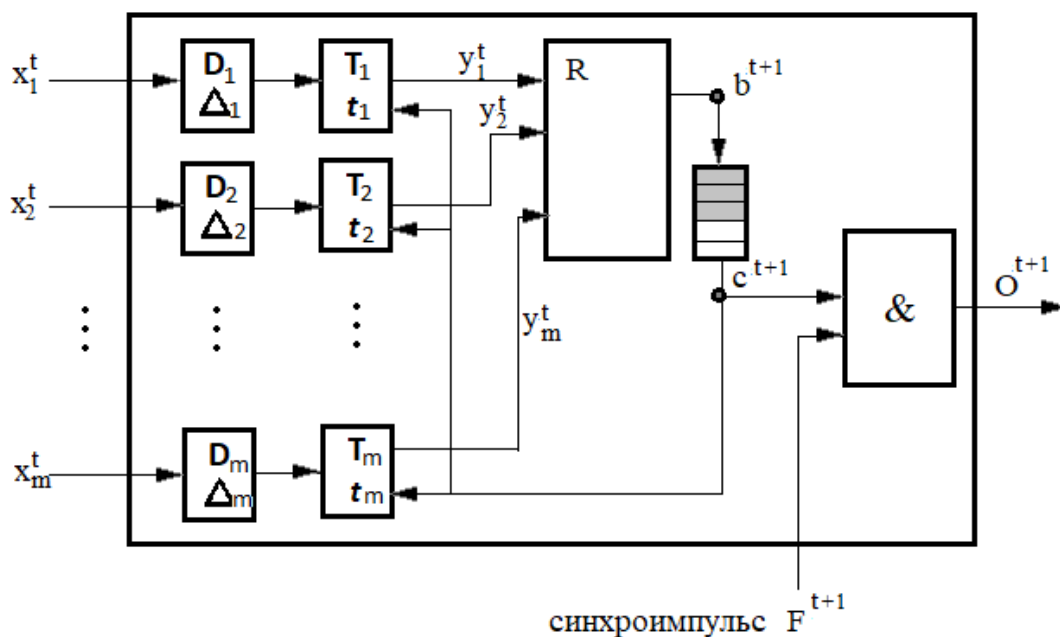


Рис. 1.11. Структура нейрона Жданова 2-го типа, которая специфицировано учитывает синаптические задержки и временные характеристики сигналов.

Здесь:

- Входной вектор $X^t = (x_1^t, x_2^t, \dots, x_m^t)$.
- Вектор проверки $Y^t = (y_1^t, y_2^t, \dots, y_m^t)$.
- Блок правила «R» - Если комбинации сигналов удовлетворяет правилу R (например, логическое И), то он выдаст сигнал на следующем такте на счётчик L через точку b.
- Блок синхронизации «&».
- Δ_i – допустимая длительность входного импульса.
- t_i – допустимая задержка поступления входного импульса.

Задача: создать структуру нейрона 2-го типа для определения последовательности двух сигналов.

Решение: Возьмем модель нейрона 2-го типа, у которого есть 2 входа (рис. 1.12). Как сказано выше, если значения времени задержки первого входа больше чем значение задержки второго входа, то нейрон будет сработать, когда сигналы x_1, x_2 последовательно ($x_1 \Rightarrow x_2$) поступают на входы нейрона. Очевидно, что если эти сигналы одновременно поступают на входы нейрона, то нейрон не сработает.

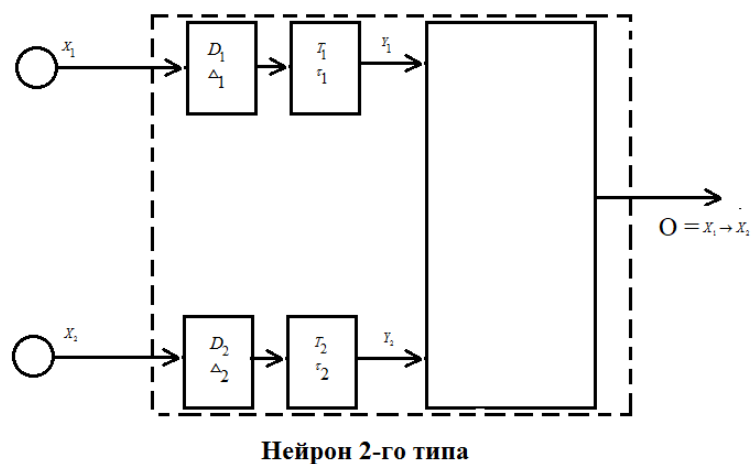


Рис. 1.12. Структура нейрона 2-го типа с двумя входами для распознавания последовательности двух сигналов [47].

Чтобы получить на выходе обученного нейрона единичный сигнал в момент t , сигналы должны поступить в интервалы времени: $d_i = [t - \tau - \Delta_i; t - 1 - \Delta_i]$ и $\cap d_i \neq \emptyset$. Тогда для решения этой задачи возьмем нейрон 2-го типа с заданными задержками $\Delta_1 = 2, \tau = 1, \Delta_2 = 1, \tau_2 = 1$. С учетом задержек и длительностей, мы получим $d_1 = [t_1 - 3; t_1 - 3]$ и $d_2 = [t_2 - 2; t_2 - 2]$. Очевидно, чтобы получить $\cap d_i \neq \emptyset$, сигнал x_1 должен поступить раньше на 1 такт, чем сигнал x_2 . Это значит, что этот нейрон можно использовать для распознавания последовательности $x_1 \Rightarrow x_2$. Пример показан на временной диаграмме (рис. 1.13):

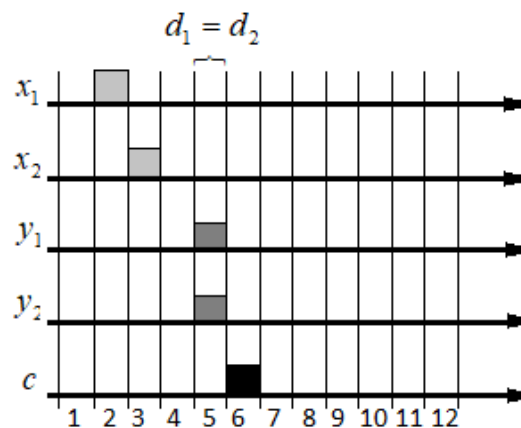


Рис. 1.13. Временная диаграмма импульсов.

Видно, что через 3 такта через блок «правила» R проходит сигнал, который через точку «с» выходит из нейрона. Тогда можно говорить о том, что нейрон сработал и распознал образ последовательности $x_1 \Rightarrow x_2$. В других случаях (рис. 1.14), если эти сигналы одновременно или последовательно $x_2 \Rightarrow x_1$ поступают на нейрон, то нейрон с заданными параметрами не сработает, потому что вектор $Y = (y_1, y_2)$ не удовлетворяет правилу R (логическое «И»).

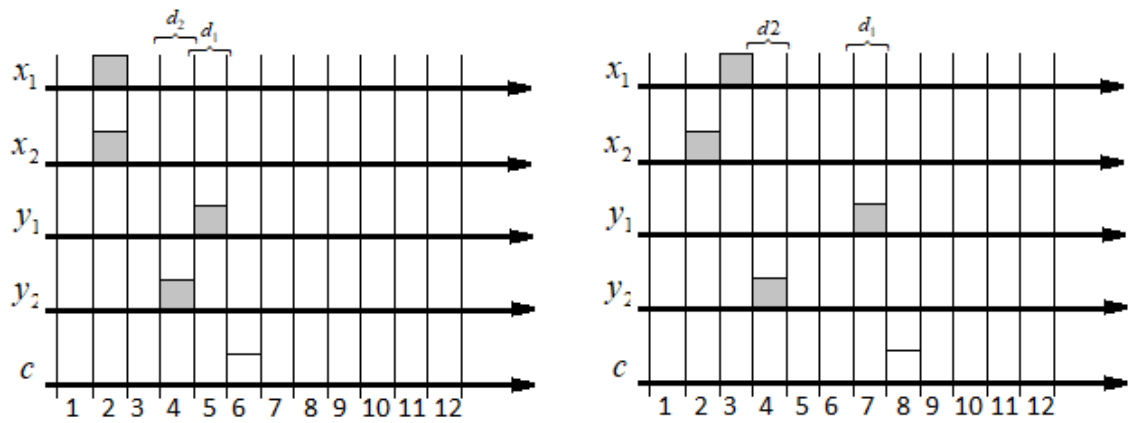


Рис. 1.14. Временная диаграмма импульсов.

Нейросетевая структура очереди. Показано, что нейрон 2-го типа является рекомпозицией подсистем нейронов 1-го типа [47]. Поэтому мы можем создать подсети на основе нейронов 1-го типа для распознавания образа очереди ($x_1 \Rightarrow x_2$). Структура очереди создана следующим способом:

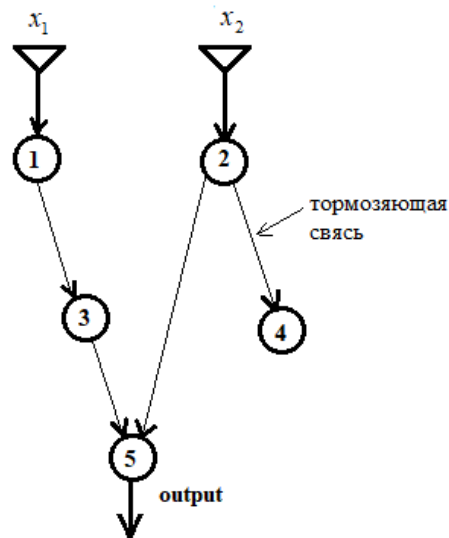


Рис. 1.15. Структура нейрона 2-го типа с двумя входами для распознавания последовательности двух сигналов [46].

В схеме нейрона 1-го типа из элемента G выдается «отключающий» сигнал Z, который по нашему мнению, выполняет функции возвратной коллатерали (рис. 1.15), блокируя боковые пути распространения возбуждения. Это значит, что сигналы направляются по определенному пути. В этой структуре «отключающий»

сигнал используется для фильтрации временных комбинаций сигналов. В этом случае структура очереди играет роль фильтра последовательности сигналов ($x_1 \Rightarrow x_2$). Временная диаграмма сигналов подсети этой структуры изображена на рис. 1.16:

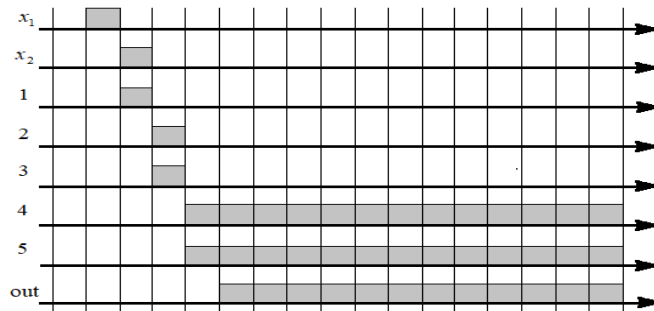


Рис. 1.16. Временная диаграмма импульсов в случае $x_1 \Rightarrow x_2$ [47].

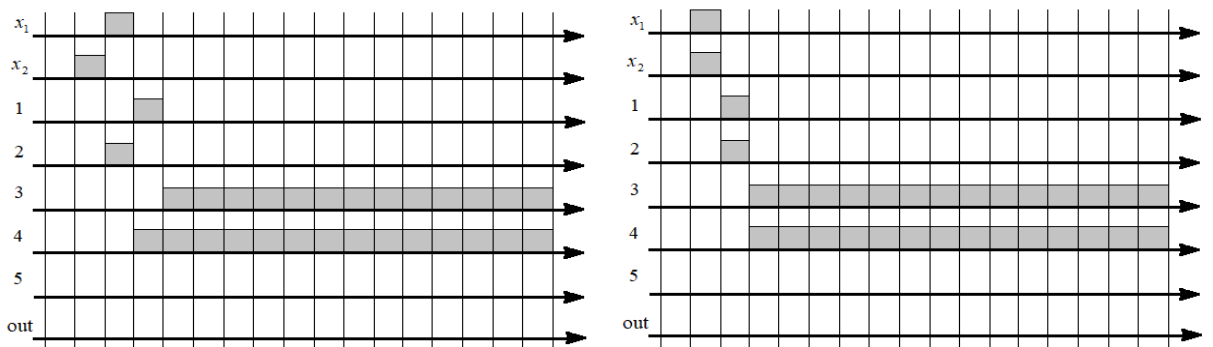


Рис. 1.17. Временная диаграмма импульсов в других случаях [47].

Вывод. В этом примере мы представили разработанные нами две элементарные структуры фильтров, которые часто использованы для построения опорных структур нейроподобных сетей. В качестве фильтров карты признаков будет построены более значимые для системы распознавания.

1.3. Описание самоорганизующейся системы формирования и распознавания образов ФРО

Синтез подсистемы формирования и распознавания образов (ФРО)

Нейронная сеть подсистемы ФРО должна уметь обнаруживать закономерные события в потоке информации, поступающей на входе ФРО из блока бинарных датчиков. Чтобы решать эту задачу, мы создали несколько алгоритмов для нахождения корреляции входных сигналов. В результате анализа этих алгоритмов нами выведено правило для автоматического создания новых нейронов и структуры в нейронной сети ФРО.

Процесс обучения нейроноподобной сети является процессом генерации структуры сети с последующим обучением нейронов [46,47]. Это важное свойство нейроноподобной сети, потому что при нахождении новых знаний, подсистема ФРО может формировать новые образы, чему соответствует появление новых нейронов в подсистеме ФРО. Как известно, когда строится самообучаемая адаптивная система, то она обязательно требует некоторой определенной априорной информации. Поэтому, в подсистеме самообучаемого блока ФРО требуется алгоритм формирования новых образов и новых нейронов.

Нейроноподобная сеть ААУ отличается от традиционной ИНС тем, что структура нейроноподобной сети нерегулярная, как это можно видеть и в биологической сети. Фрагмент структуры нейроноподобной сети изображен на рисунке 1.18.

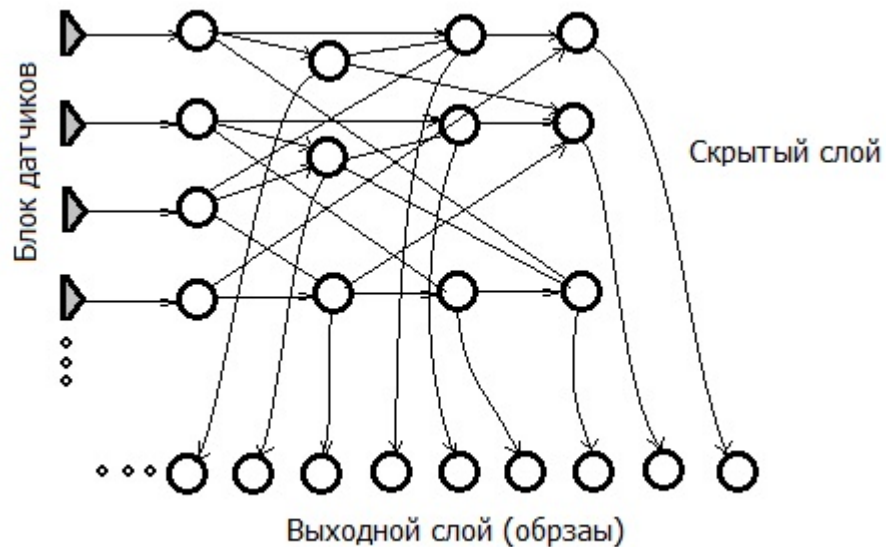


Рис. 1.18. Фрагмент нейроподобной сети.

Нейроподобная сеть подсистемы ФРО состоит из блока датчиков, входного слоя, скрытого слоя и выходного слоя. Блок датчиков составляет входные сигналы для этой сети, генерируя бинарные векторы. Например, вектор $X = (0,1,0,1,1, \dots)$ - карта образа, которая отражает некоторый прообраз, который часто появляется на входе сети. Через скрытый слой образуется вектор возбуждения нейронов, передаваемых на выходной слой подсистемы ФРО.

Задача автоматического «выращивания» сетей из нейронов в системе ААУ, является актуальной для этого метода, поскольку структуру этих нерегулярных сетей нужно формировать индивидуально для каждого конкретного объекта управления, и это является одним из уровней адаптивности таких систем. В группе Жданова А.А. рассматривалось несколько подходов к построению алгоритмов автоматического формирования сетей. Это подход на основе генетических алгоритмов [5], и подход на основе матрицы совпадений срабатываний 2-х нейронов [50,51].

В работах [50,51] используется матрица событий для накопления статистики совпадения срабатываний 2-х активных нейронов. За счет этого анализа, новый нейрон создается над теми из двух нейронов, число совпадения срабатываний которых достаточно большое. Подход показал свою

эффективность, его ограничения в том, что может совпадать активность более, чем только 2-х нейронов, и возникают нежелательные эффекты, например: если «на вход которой последовательно подавался вектор {111}, то это приводило к ее бесконечному воспроизводству» [50] и количество сгенерированных нейронов получалось слишком большое. Для устранения этих недостатков, мы разработали новый алгоритм роста нейроноподобных сетей на основе дерева событий.

«Дерево событий». Основная задача состоит в том, чтобы найти алгоритм роста сети, который будет автоматически добавлять в сеть блока ФРО нейроны, которые с наибольшей вероятностью будут использованы в данной системе. В этой работе мы представляем алгоритм роста нейроноподобной сети на основе дерева событий. Дерево событий [49] (ДС) (рис. 1.19) используется для накопления статистики совпадения срабатываний нескольких активных нейронов, чтобы находить закономерные комбинации образов.

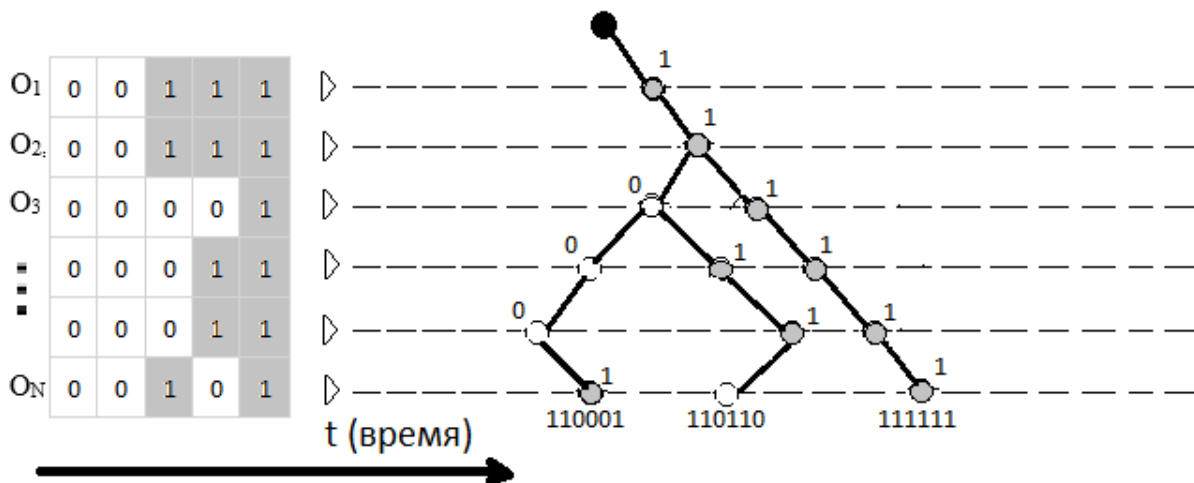


Рис. 1.19. Схема дерева событий (ДС).

По нашему убеждению, каждый нейрон соответствует одному образу, которому можно присвоить порядковый номер. Каждый порядковый номер соответствует также номеру уровня ДС.

Определение 1: *выходной вектор* $O(t)$ *подсистемы ФРО* – это вектор, образованный из выходных сигналов нейронов выходного слоя образов.

Определение 2: *вектор ситуации* $S(t)$ *подсистемы ФРО* – это вектор, который образуется из всех выходных сигналов нейронов скрытого слоя.

Выходной вектор $O(t)$ ФРО отражает состояние распознанных образов. Пусть вектор $O(t)$ поступает на вход ДС, который выполняет процесс накопления распознанных образов и сохраняется в ДС с помощью алгоритма самоорганизующейся структуры ДС. Как сказано выше, вектор $O(t)$ подсистемы ФРО образуется из всех выходных сигналов нейронов выходного слоя, поэтому элементы вектора $O(t)$ состоят из элементов множества $\{0,1\}$ - $O(t) = (1,0,0,1,1,0,\dots,0,1)$. Опишем теперь алгоритм построения ДС по последовательному просмотру компонент вектора $O(t) = (1,0,0,1,1,0,\dots,0,1)$:

- 1) если текущий элемент вектора $O(t)$ равен «1», то ДС строит правый узел относительно предыдущего или увеличивает счетчик правого узла, если он уже создан;
- 2) если текущий элемент вектора $O(t)$ равен «0», то ДС строит левый узел относительно предыдущего или увеличивает счетчик левого узла, если он уже создан.

Например, см. рис. 1.20.

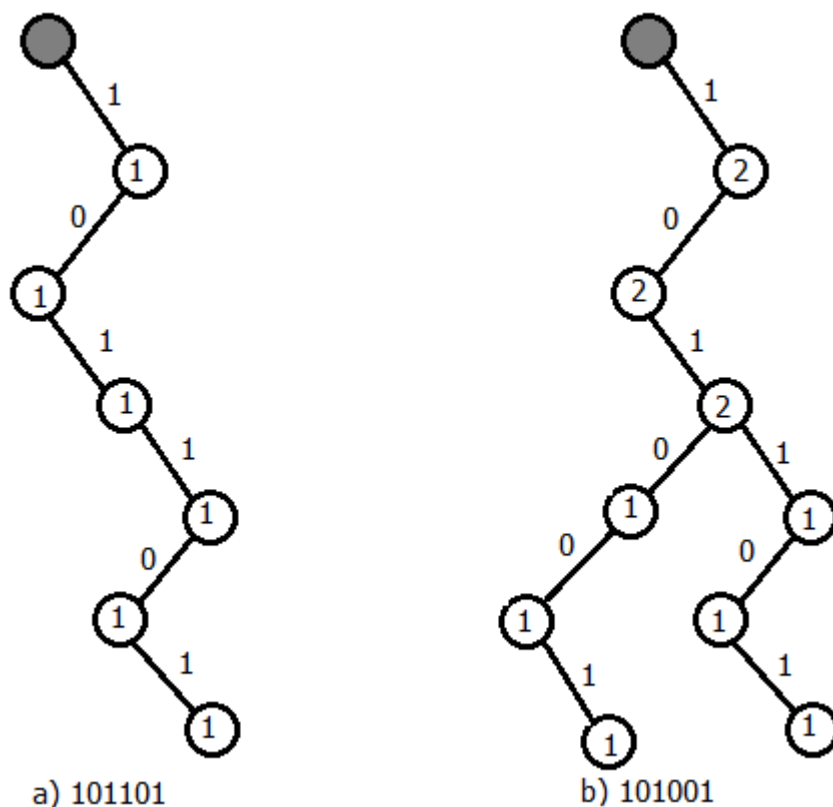
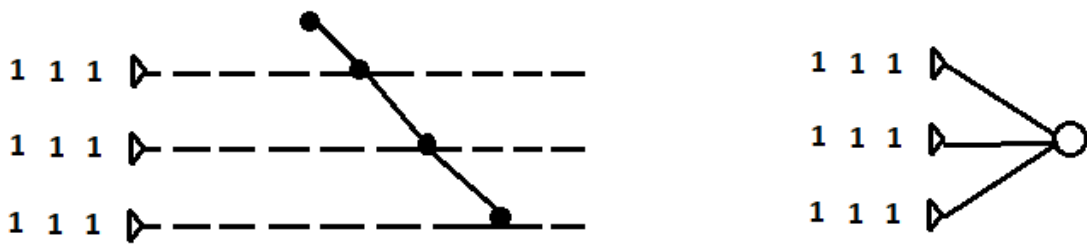


Рис. 1.20. Примеры построения ДС.

В каждый такт времени ведется учет совпадений сигналов от всех нейронов выходного слоя подсистемы ФРО. Если число совпадений больше, чем заданное число L , которое выбирается достаточно большим, чтобы обеспечить неслучайное событие совпадения нейронов, то создается новый нейрон в подсистеме ФРО от совпадающих активных нейронов [49].

На рис 1.21 изображены ДС и сеть ФРО, где вектор (1,1,1) последовательно подавался на вход сети ФРО, в результате только 1 образ был сформирован.

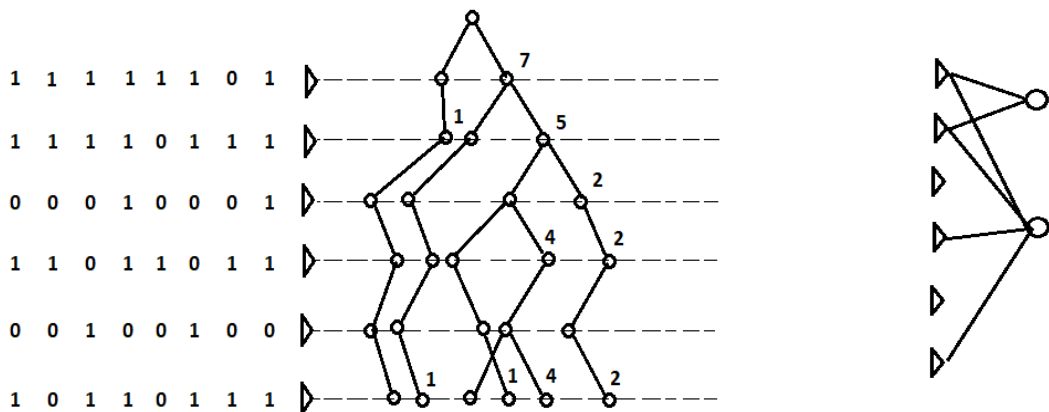


а) дерево событий

б) нейронная сеть

Рис. 1.21. Формирование образов сети ФРО при подаче вектора {111}.

Второй эксперимент: в сети ФРО тоже успешно обнаружены и сформированы образы, при подаче информации от прообразов с помехами (рис.1.22).



а) Дерево событий

б) Нейронная сеть

Рис. 1.22. Формирование образов сети ФРО при подаче векторов с зашумлением прообразов.

Бинарное дерево так же используется для построения модели подсистемы формирования и распознавания образов мобильного робота [51]. В работе [51], подсистемы ФРО с использованием модифицированного стекового алгоритма поиска [51] в бинарном дереве, которое используется для структурирования входного вектора (статического образа). Порожденный узел этого дерева при поступлении информация прообраза, являющийся последним называется «терминальным» и содержит в себе идентификатор образа. Это значит,

количество сформированных образов равно количеству последних узлов дерева. Здесь система формирования и распознавания образов работает как процесс поиска в дереве с помощью модифицированного стекового алгоритма [51], чтобы найти узел с максимальным значением меры близости [51].

1.4. Языковые явления и основная нейроподобная конструкция, порождающая язык в нейроподобной системе ААУ

Помехоустойчивость нейрона

Вернемся к модели нейрона 1-го типа. При наблюдении прообраза, который состоит из некоторых бинарных компонентов $(1,0,1,\dots,1)$, нейрон должен уметь распознавать этот образ, если в прообразе не хватает некоторого числа единичных сигналов. Знаем, что при распознавании образа блок проверки «структурного правила» R проверяет входной вектор, и, если входной вектор удовлетворяет правилу R , то счетчик L увеличивает число N на 1. С ростом числа N уменьшается вероятность ложной тревоги, и мы можем уменьшать требования к полноте входного вектора. В блоке R сделаем пороговую функцию, чтобы реализовать свойство помехоустойчивости нейрона.

$$\rho = h / m \in (0,1). \quad (1.5)$$

Здесь h – количество единицы входного вектора, m – размерность вектора.

Тогда порог ρ будем уменьшать в зависимости от числа N , эта зависимость может быть следующего вида (рис. 1.23).

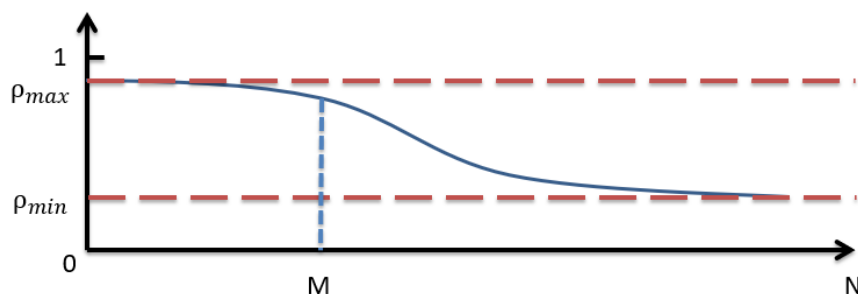


Рис. 1.23 Снижение порога нейрона.

Свойство помехоустойчивости можно использовать для решения следующих задач:

- распознавание образа в условиях помех и при неполной информации;
- порождение ассоциативных образов;
- порождение языка.

Нейроподобные конструкции, порождающие язык в системе ААУ

Обратим внимание на способность нейроподобной системы ФРО распознавать образы в условиях помех, и заметим, что с помощью этой способности нейрон можно использоваться для порождения ассоциативно связанных образов. Возьмем конструкцию, которая состоит из трех нейронов n_a , n_b , n_c (рис. 1.24). Нейрон n_a обучен и распознает образ A , а нейрон n_b обучен и распознает образ B .

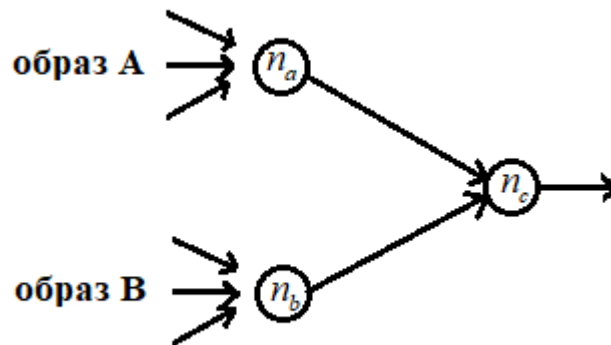


Рис. 1.24. Конструкция ассоциативно обучения.

Если прообразы образов A и B одновременно несколько раз поступают на обученные нейроны n_a и n_b , то через некоторое время это приводит к обучению нейрона n_c . Очевидно, что если нейрон n_c обучен, то он может срабатывать при наличии обоих образов. Если через некоторое время порог нейрона n_c упадет до величины менее 50%, то этот нейрон сможет распознать свой образ даже при

поступлении только одного из этих двух прообразов. При этом возникает явление идентификации, которая является основой языка.

Можно видеть, что явление идентификации происходит в УС при обучении некоторых воздействий (*условие & действие*) → *следствие*. Это значит, что процесс идентификации возможен не только для объектов, а также для действий – языковых действий.

$$\begin{array}{ccc} \text{идентифкатор} & \text{идентифкатор} & \text{идентифкатор} \\ (\text{Условие} & \& \text{действие}) & \Rightarrow & \text{следствие} \end{array}$$

Синтез самообучаемой системы распознавания звуков.

Как было сказано, подсистема ФРО в системе ААУ является самообучающейся системой, процесс обучения у которой является процессом генерации структур сети и их последующего обучения. Если какой-то звук много раз поступает на вход подсистемы ФРО, то подсистема ФРО анализирует поток звуковых сигналов и пытается найти корреляцию в этом потоке [47]. С помощью алгоритмов роста сети в подсистеме ФРО идет процесс генерации структуры сети. В результате этого алгоритма в сети достраиваются нейроны, которые соответствуют тем наиболее часто повторяющимся звукам, которые можно считать прообразами слов [47] (паттернами). В отличие от других методов, наша нейронная сеть обучается с небольшим количеством данных, при обучении новый образ добавляет в сети новую нейронную структуру. При этом не требуется глобального переобучения, как в ИНС.

Речевые сигналы характеризуются тремя атрибутами – амплитудой, частотой и фазой, которые накапливаются во времени, представляя собой пространственно-временные образы [47]. Смоделируем процесс распознавания речи, используя рациональные представления об основных механизмах этого процесса. К примеру, сначала сеть обучается распознавать простые звуки, а

потом более сложные слова, которые состоят из последовательности простых звуков (рис. 1.25).

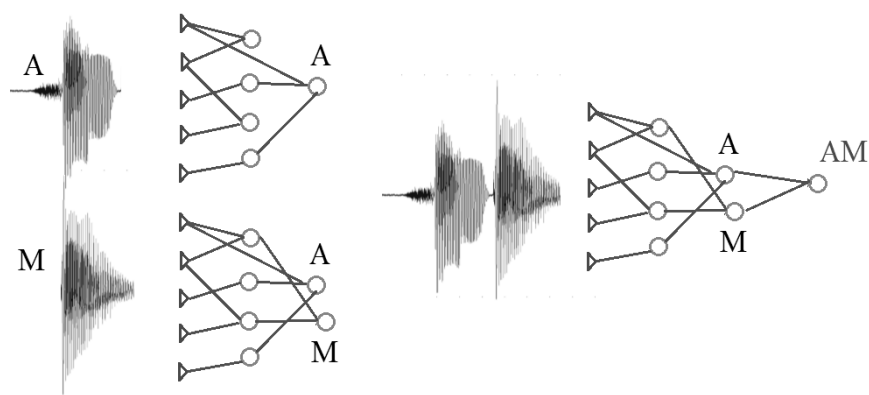


Рис. 1.25. Синтез самообучаемой подсистемы распознавания звуковых сигналов [47].

1.5. Описание модели формирования базы знания на основе нейроподобных сетей

По методике ААУ, основными целевыми функциями УС автономного ОУ являются:

- выживание ОУ;
- накопление знаний о системе «среда – ОУ – ОС».

Элементарное знание в системе ААУ состоит из трех основных элементов. Первый элемент знания – это «условие», оно представлено логическим выражением из распознанных образов в текущий момент. Второй элемент знания состоит из «действия». Третий элемент знания – «результат» состоит из распознанных образов, которые получены в результате совершения «действия». Таким образом, «тройка» - знаний в системе ААУ записывается следующим логическим выражением-импликацией:

(Условие & действие) → результат,

или в следующем виде:

$$(O_i(t-2) \& D_j(t-1)) \rightarrow O_z(t),$$

где: $O_i(t-2)$ - образы «Условия», $D_j(t-1)$ - образы «действий» и $O_z(t)$ образы «результатов».

Упорядоченное определенным способом множество элементарных знаний составляет Базу Знаний (БЗ) системы ААУ. Одним из способов построения базы знаний является табличный способ. Это самый простой способ построения базы знаний, но он очень удобен и его легко реализовать во многих прикладных системах ААУ. Табличный способ построения БЗ наиболее удобен в самом простом случае, когда:

- в каждый момент времени « $t-2$ » подсистема ФРО распознает не более чем один образ из множества $O_i(t-2)$, где $i = 1, 2, \dots, N$ (N - количество образов «условия»);
- в момент « $t-1$ » блок принятия решения выбирает только одно действие из множества «действий» $D_j(t-1)$, где $j = 1, 2, \dots, M$ (M - количество действий);
- в момент « t », в результате мы получим множество распознанных образов, которые являются результатами выбранного действия.

Если знания удовлетворяют перечисленным условиям, то мы создаем прямоугольную таблицу размера $N \times M$ (N - количество образов «условия» и M - количество действий) (таб. 1.1). Тогда в ячейке с индексом (i, j) мы можем записать номер образа результата и получим «тройку (i, j, z) », соответствующей событиям $(O_i(t-2) \& D_j(t-1)) \rightarrow O_z(t)$. Назначим некоторое пороговое значение L . Если «тройка (i, j, z) » встречается в предыстории больше чем L раз, то «тройка (i, j, z) » является неслучайным событием, и можем записать ее в таблицу базы знаний.

		Номер <i>i</i> распознанного образа условия					
		1	2	...	<i>i</i>	...	N
Номер <i>j</i> образа действия	1						
	2						
	⋮						
	<i>j</i>				z		
	⋮						
	M						

Таб. 1.1. Представление табличной Базы знаний.

Базу знаний также можно реализовать нейроподобным способом на основе нейронов 1-го типа. Нейрон 1-го типа обладает свойствами: самостоятельно обнаруживает закономерные бинарные сигналы прообраза и затем распознает этот образ при неполной информации. Тогда, подавая на вход нейрона информацию о компонентах элементарного знания $(O_i(t-2) \& D_j(t-1)) \rightarrow O_z(t)$, взятых из предыстории, можно заставить нейрон сформировать образ такого события – элементарного знания. Соответствующая конструкция на основе нейронов изображена на рис. 1.26.

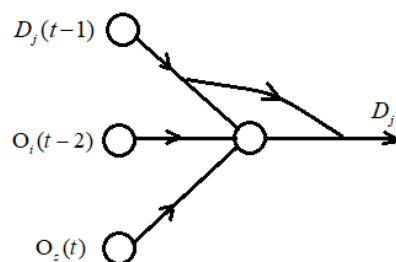


Рис. 1.26. Элементарная конструкция нейронов базы знаний.

Для обучения БЗ используется случайный или любой другой выбор действия с помощью замыкания входа $D_j(t-1)$ на выход $D_j(t-1)$, поэтому обучение БЗ происходит в результате воздействий на среду. Если $(O_i(t-2) \& D_j(t-1)) \rightarrow O_z(t)$ много раз повторяется, то нейрон БЗ обучится и будет сформировано новое элементарное знание. Через некоторое число повторений актов наблюдения, порог нейрона снижается ниже, чем величина $(n-1)/n$ (где n – количество входов нейрона БЗ). Тогда нейрон БЗ может сработать при отсутствии одного из входов нейрона БЗ. Например, если подсистема ФРО распознала образ «условия» $O_i(t-2)$, и мы подадим пробный сигнал образа «результата» $O_z(t)$, и если найдется нейрон, который обучился этой тройке (элементарному знанию), то он сработает и выдаст на выход сигнал D_j , который можно направлять в блок исполняющего органа. Сигнал D_j сообщает, что, в текущих условиях именно с помощью этого действия можно получить желаемый результат. Если ни один нейрон в БЗ не откликнулся, значит, в текущих условиях искомого результата получить нельзя. Полная база знаний построится в виде 3-мерной матрицы нейронов (рис. 1.27).

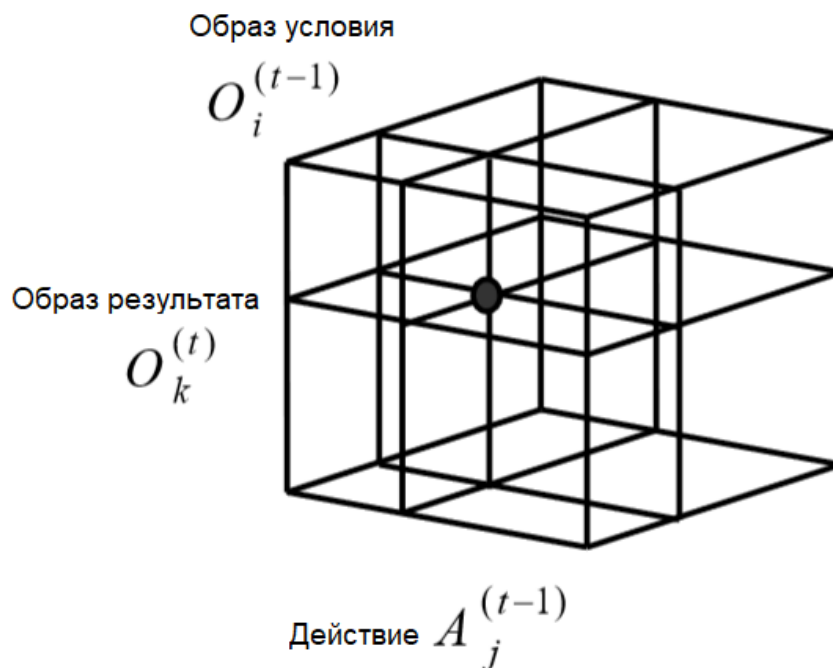


Рис. 1.27. 3-мерная матрица нейронов базы знаний.

Методы реализации базы знаний, описанные выше, имеют некоторые недостатки:

- когда подсистема ФРО распознала несколько образов «условия», и случайно выбирается один образ из распознанных образов, тогда не всегда мы получим правильное действие из БЗ;
- трудно выбрать и записать в БЗ при распознавании нескольких образов результатов. Либо случайно выбрать из распознанных образов, либо усредняем оценки образов.

Чтобы устранять эти неудобства, представляем новый способ для построения БЗ на основе нейронов разных типов. Возьмём элементарную конструкцию узла базы знаний и делаем модификацию этой структуры:

- заменим нейрон результата из нейрона I-го типа на нейрон III-го типа;
- добавим связь «переключения» от нейрона «условия» в нейрон «результата».

Тогда процесс обучения базы знаний состоит из двух этапов:

- обучение нейрона «результата»;
- обучение нейрона базы знаний.

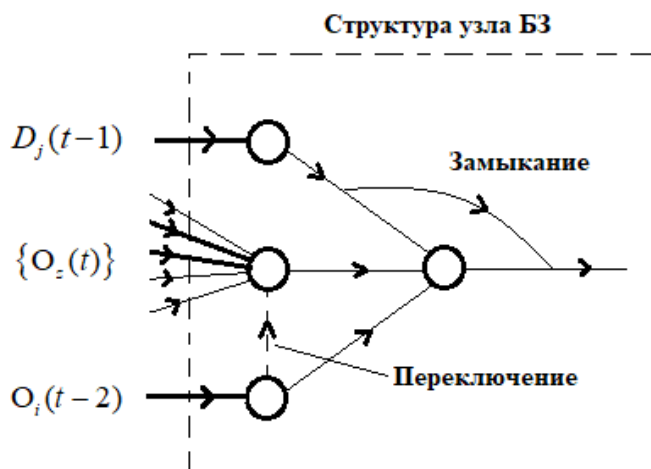


Рис 1.27. Новая конструкция узла базы знаний.

Теперь рассмотрим на диаграмме потоки информации, которые поступают на вход базы знаний:

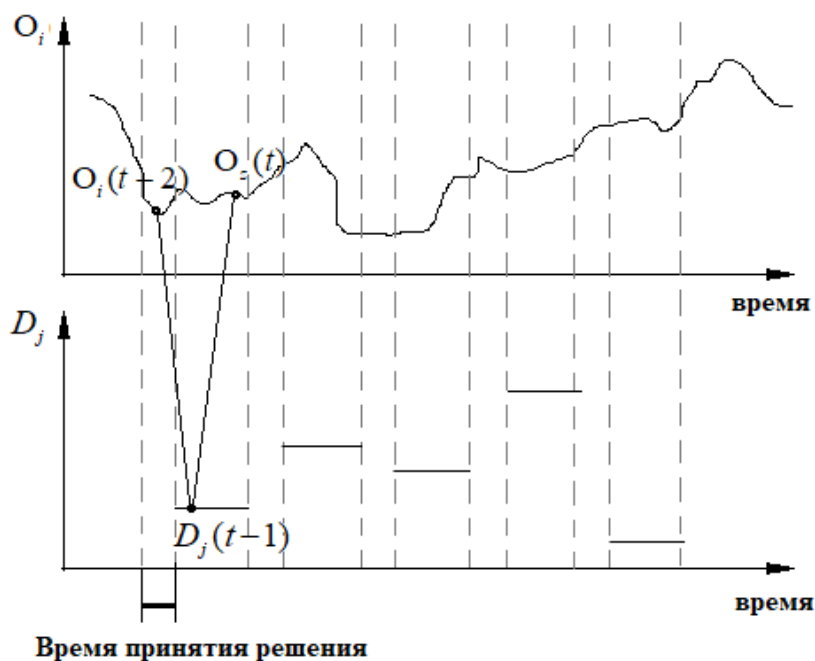


Рис 1.28. Временные потоки информации системы управления. O_i – образы текущей ситуации, D_j – действия, совершаемые системой управления.

Мы заметим, что в системе ААУ число одновременно распознанных образов не всегда равно 1, а может быть 2 или больше. Поэтому мы заменили нейрон «результата» на нейрон III-го типа, чтобы он мог обучиться на полной информации из множества распознанных образов, которые образуют текущую ситуацию.

Новая конструкция позволяет БЗ обучаться в реальном времени, и База знаний должна обнаружить и обучиться закономерным знаниям $(O_i(t-2) \& D_j(t-1)) \rightarrow O_z(t)$. В фазе обучения на вход базы знаний поступают сигналы множества распознанных образов $O_i(t-2)$. В следующий момент «t-1» нейроны условия переключают нейроны «результатов» из режима «использования» в режим «обучения» и УС случайно совершает действие $D_j(t-1)$. В момент «t», сигналы множества $\{O_z(t)\}$ распознанных образов, которые являются результатами действия $D_j(t-1)$, поступают на вход нейронов в режиме «обучения». Нейроны обучаются этим сигналам изменением значения своих весов, выдавать на выход сигнал «1» может только обученный нейрон. Выходной сигнал нейрона III-го типа определяется соотношением:

$$O_w^t = \begin{cases} 1, & \text{if } \sum_{i=1}^m x_i^t w_i^t \geq \theta \\ 0, & \text{in other cases} \end{cases} \quad (1.6)$$

Где x_i^t - i-й элемент входного вектора, $x_i^t \in \{0,1\}$, w_i^t - вес i-ой связи нейрона и θ пороговое значение, $0 < \theta < 1$.

В начальный момент $t=0$, нейрон иницируется с одинаковыми весами, которые $\ll 1$ (рис. 1.28). Эти веса определяются по формуле:

$$w_i^0 = \frac{\delta}{m}, \quad 0 < \delta < \theta \quad (1.7)$$

Где δ - свободный параметр, m - размерность входного вектора.

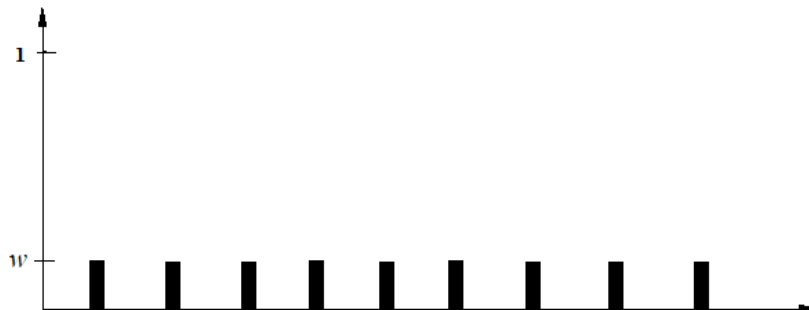


Рис. 1.29. Инициализация значений весов нейрона III-го типа одинаковыми весами.

Процесс обучения производится при поступлении входного вектора с изменением весов нейрона. Новое значение веса определяется по формуле:

$$w_i^{t+1} = w_i^t + dw_i^{t+1} \quad (1.8)$$

Где приращение dw_i^{t+1} функционально зависит от поступления входного вектора и предыдущих значений весов нейрона - $dw_i^{t+1} = f(\vec{w}^t, \vec{x}^t)$.

$$dw_i = f_i(\vec{w}^t, \vec{x}^t) = x_i^t w_i^t \left(1 - \sum_{j=1}^m w_j^t\right) + q w_i^t (x_i^t - 1) \quad (1.9)$$

Где q - свободный параметра $0 < q < 1/m$, который должен быть таким, чтобы уравновесить положительное и отрицательное приращения.

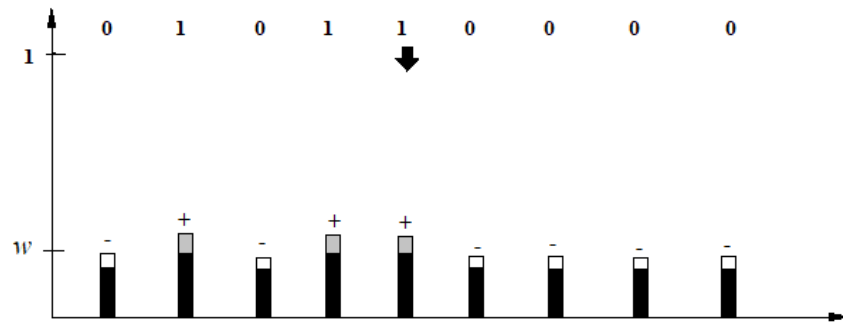


Рис. 1.30. Процесс обучения нейрона III-го типа.

Когда будет получен первый сигнал $O_w^t = 1$, мы можем сказать, что нейрон обучен и образ сформирован. Значения весов обозначают вероятности появлений соответствующих элементов входного вектора.

После того, как нейрон «результатов» обучен, этот нейрон может выдавать на выход сигнал $O_w^t = 1$ на нейрон БЗ. Обучение БЗ на основе нейронов может происходить в течение всей жизни системы и параллельно с принятием решений. В таком варианте реализации БЗ, если на вход базы знаний поступает вектора текущей ситуации, который состоит из распознанных образов, то в БЗ найдутся некоторые обученные нейроны с разными образами действий. Результаты у этих нейронов можно вычислить через нейрон «результата» этих узлов следующим способом:

$$S_j = \sum_{i=1}^m o_i w_{ji} , \quad (1.10)$$

где S_j - эмоциональная оценка результата при совершении действия A_j , o_i - эмоциональная оценка образа O_i , w_{ji} - значение i -го веса j -го нейрона БЗ.

В результате мы получим список $\{S_j\}$, блок принятия решения будет выбирать действие с максимальной оценкой и отправлять его в блок исполняющего органа.

1.6. Выводы по главе 1

В этой главе мы представили две новые элементарные нейроноподобные структуры, которые способны к построению нейроноподобных сетей для распознавания статических и динамических образов, в частности это система распознавания звуков (их применение будет описано во 2-й главе). Новый алгоритм роста нейроноподобных сетей ФРО (дерево событий ДС) был успешно реализован и применен в системе управления ААУ. Алгоритм ДС построения ФРО устраняет некоторые недостатки старых алгоритмов. Новый алгоритм роста сети обладает следующими преимуществами:

- уменьшение количества нейронов в подсистеме ФРО, за счет чего уменьшается объем вычисления;
- автоматически выращиваемая таким способом нейроноподобная сеть подсистемы ФРО хорошо обнаруживает и формирует образы закономерных прообразов.

ГЛАВА 2. Алгоритмы построения самообучаемой системы распознавания статических образов и системы распознавания звуков на основе адаптивных нейроподобных элементов (нейронов)

Подсистема ФРО в системах ААУ решает две задачи: находит закономерные прообразы (формирует образы) и распознает образы. В прикладных системах ААУ в качестве подсистемы ФРО может выступать любая подходящая самообучаемая система распознавания образов (или не самообучаемая, если этого достаточно в конкретном случае). Это могут быть вероятностные, геометрические (в частности ИНС), логические, структурные или комбинированные системы распознавания образов.

Прикладными примерами могут служить, например:

- **системы распознавания статических образов**, такие как: система распознавания марок машины, система распознавания дорожных знаков и т.д. Входные данные от прообразов, которые не изменяют свои свойства во времени, приходят здесь из видеокамеры или других видов сенсоров;
- **системы распознавания динамических образов**, такие как: система распознавания траекторий движения подвижных объектов, система распознавания звуков и т.д. Входные данные для прообразов являются здесь пространственно-временными.

Архитектура программы для решения этой задачи состоит из следующих блоков (рис. 2.1.).



Рис. 2.1. Архитектура самоорганизующейся системы ФРО.

Здесь:

- рецепторное поле состоит из набора датчиков, которые преобразуют входные данные в бинарный вид;
- блок модификации сети получает информацию о состояниях нейронной сети и анализирует эти состояния разными методами. В результате анализа блок модификации возвращает новую структуру с новыми нейронами или сообщение о списке нейронов, которые будут удаляться;
- выходные нейроны показывают состояние всех распознанных или не распознанных образов.

2.1. Алгоритмы динамического построения сети для формирования и распознавания статических образов

Исходные данные для визуальных прообразов, которые поступают на вход системы распознавания, являются изображениями размера $N \times M$. Нейроны Жданова в используемом наиболее простом их варианте работают с бинарными сигналами $\{0,1\}$. Поэтому надо построить карту признаков из исходных матриц данных в бинарном виде. Эта предобработка исходных данных состоит из следующих этапов:

- нормализация – это замена исходной матрицы данных на скалярную матрицу;
- переход из цветового пространства RGB в цветное пространство L^*a^*b Модели CIELAB [33];
- преобразование в бинарный вид.

Нормализация - это задача замены исходной матрицы с размером $H \times W$ в N скалярных матриц с разрешением $H^n \times W^n$ (нормальный размер). Число N скалярных матриц удовлетворяет условию $0 < N < (H \times W) / (H^n \times W^n)$. В результате нормализации мы получаем N скалярных матриц, которые изображены на рисунке 2.2.

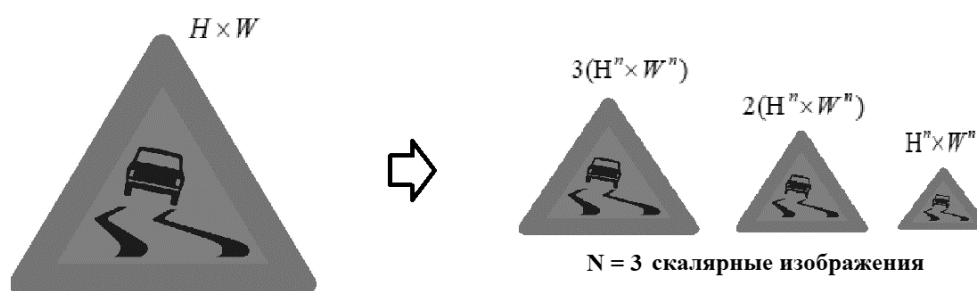


Рис. 2.2. Нормализация входных изображений.

При нормализации соотношение $c = (H \times W) / (H^n \times W^n)$ сохраняется во временной памяти или в таблице «объект – свойство» для разных целей систем распознавания. Во-первых, это соотношение используется для сравнения размеров распознанных образов. Во-вторых, это соотношение является одним из свойств образа и его можно использовать для систем распознавания со знаниями. Например, имеется система распознавания реальных объектов. Если при обучении в системе сохранены все возможные размеры одного класса и при появлении новый объект очень похож на один класс при сравнении нескольких признаков, но размер совсем отличается от обучаемых размеров этого класса, то

результат распознавания может отвергаться (например, не бывает человека с ростом 4 метра).

Переход из цветового пространства RGB в цветовое пространство L^*a^*b . Любой цвет определяется яркостью L и двумя хроматическими компонентами: a изменяется в диапазоне от зеленого до красного, b – в диапазоне от синего до желтого. Основной целью при разработке CIELAB [33] было устранение нелинейности системы RGB с точки зрения человеческого восприятия.

Часто используемая матрица перехода:

$$\begin{matrix} L \\ a \\ b \end{matrix} = \begin{matrix} \left\{ \begin{matrix} 0.299 & 0.587 & 0.114 \\ 0.5 & -0.4187 & -0.0813 \\ 0.1687 & -0.3313 & 0.5 \end{matrix} \right\} \times \begin{matrix} R \\ G \\ B \end{matrix} + \begin{matrix} 0 \\ 128 \\ 128 \end{matrix} \end{matrix}$$

В данной работе, мы используем только канал L , который является черно-белой частью картины. Известно, что значение L находится в диапазоне $[0 \div 100]$.

Преобразования в бинарный вид

С помощью пороговой функции, канал L картины легко преобразуется в бинарный вид:

$$x_{i,j} = \begin{cases} 1, & \text{если } L_{i,j} \geq \rho \\ 0, & \text{если } L_{i,j} < \rho \end{cases}$$

Где - $L_{i,j}$ значение интенсивности света в пикселе (i,j) , ρ - пороговое значение.

Обучение. Процесс обучения сети является процессом генерации структур сети и обучения нейронов. Алгоритм синтеза биоморфной нейроподобной

системы распознавания состоит из 2-х этапов. Это этап синтеза структуры подсети библиотеки фильтров для каждого поступающего образа и этап синтеза сети для распознавания образа. Структура фильтров описана в главе 1/2.1./ «Нейросетевой фильтр» [46]. Эти два этапа рекурсивно повторяются, генерируя слой за слоем, пока не будет синтезирована нейронная сеть, обеспечивающая за счет своей структуры полное распознавание всего множества заданных образов. Сгенерированная сеть обучается на каждой итерации алгоритма так, что каждый нейрон обучается отдельно. В случае необходимости дообучения уже сформированной сети, этот же алгоритм применяется для ее дотройки и дообучения. Общая схема алгоритма синтеза структуры подсети библиотеки фильтров-масок для каждого поступающего образа приведена на рис. 2.3.

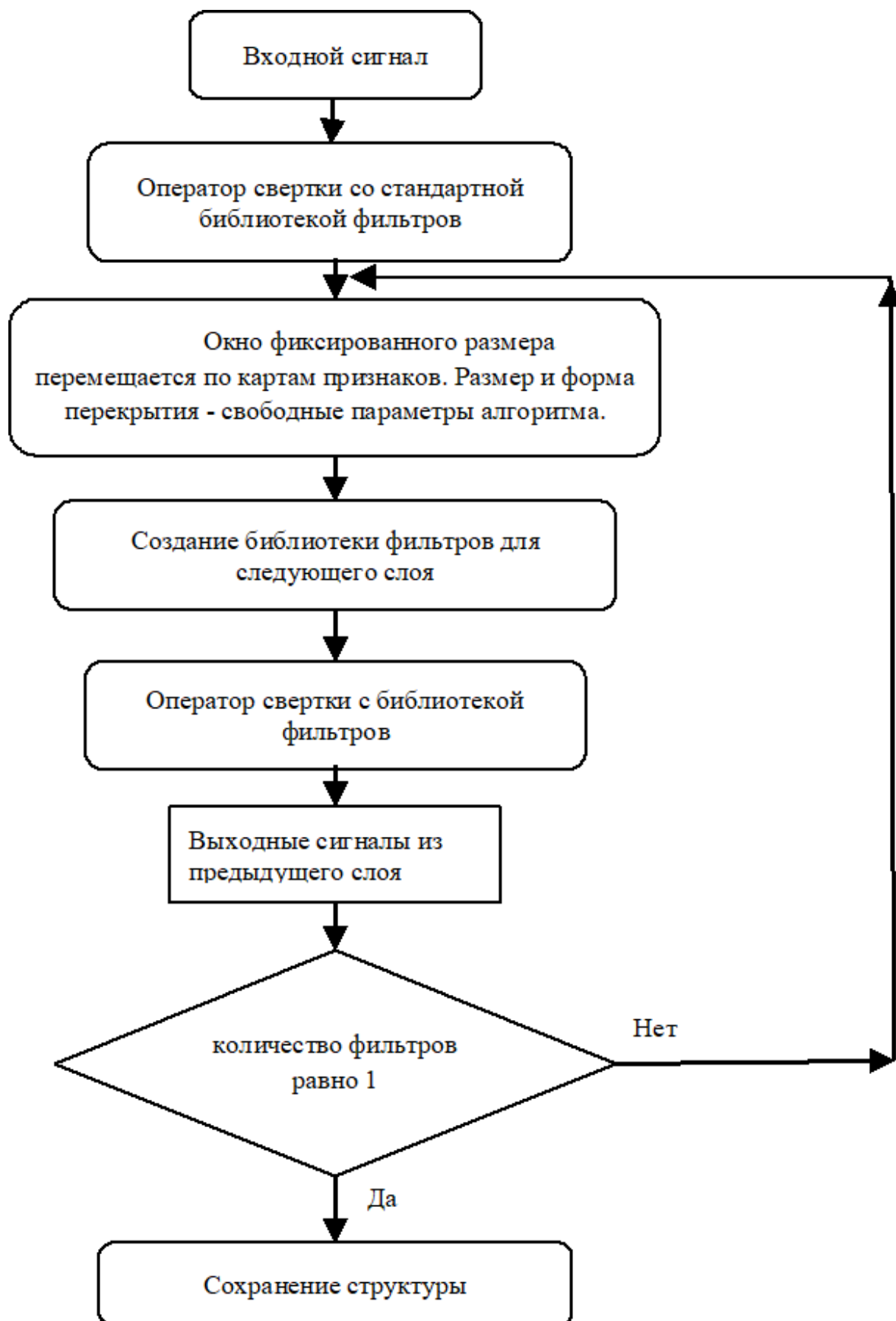


Рис. 2.3. Рекурсивный алгоритм создания библиотеки фильтров [46].

Обучение может быть как с учителем, так и без. Приведенный алгоритм может использоваться для создания самообучаемой системы распознавания образов, которая может порождать новые нейроны со структурой, которые способны отражать как наиболее вероятные, по критерию частоты появления, так и заранее заданные образы (работа не с вероятностями, а по прецедентам).

Первый слой генерируется через стандартную библиотеку фильтров, например, множество прямых линий (рис. 2.4).

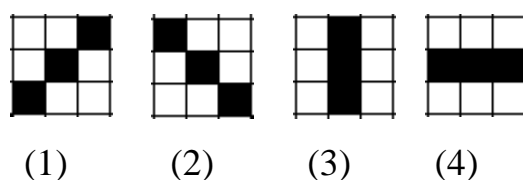


Рис. 2.4. Пример стандартной библиотеки фильтров для распознавания визуальных образов.

Тогда структура подсети для одного фильтра [46] будет следующей (рис. 2.5).

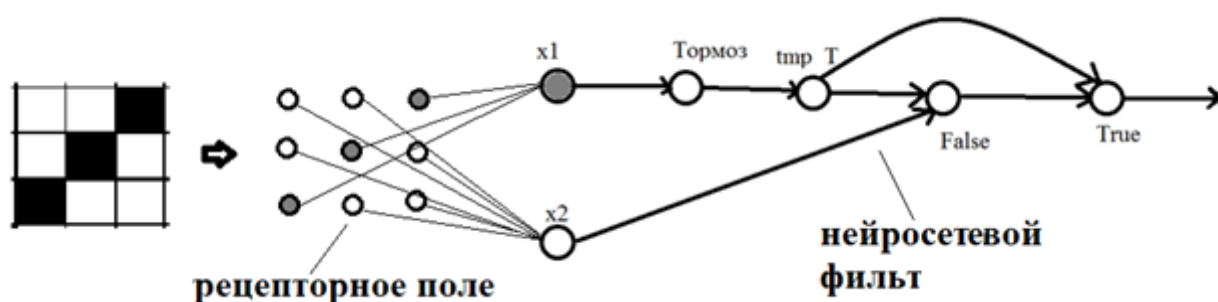


Рис. 2.5. Фрагмент подсети нейросетевого фильтра [46].

Нейрон, отвечающий за тот или иной распознанный образ, идентифицирует структурой своей подсети этот образ, тем самым соотнося его со своим идентификатором или неким семантическим описанием. Отметим, что такие

идентификаторы может назначать другая сеть, которая сформирована другой системой фильтров-масок от других рецепторов. Может существовать множество подсетей/образов, которые не имеют собственных идентификаторов. При этом уместно говорить о моделировании процесса описания образов и о речевых эффектах.

На рис. 2.6 схематически изображен пример применения приведенного алгоритма для создания библиотеки фильтров для распознавания графического образа «X» и результаты его распознавания. Как и в общей задаче распознавания произвольных образов, количество фильтров и скрытых слоев заранее неизвестно [46]. В результате была алгоритмически синтезирована структура сверточной нейронной сети для распознавания образа символа «X» в условиях помех и зашумлений. В последнем слое мы получили условные координаты каждого распознанного образа (рис. 2.6).

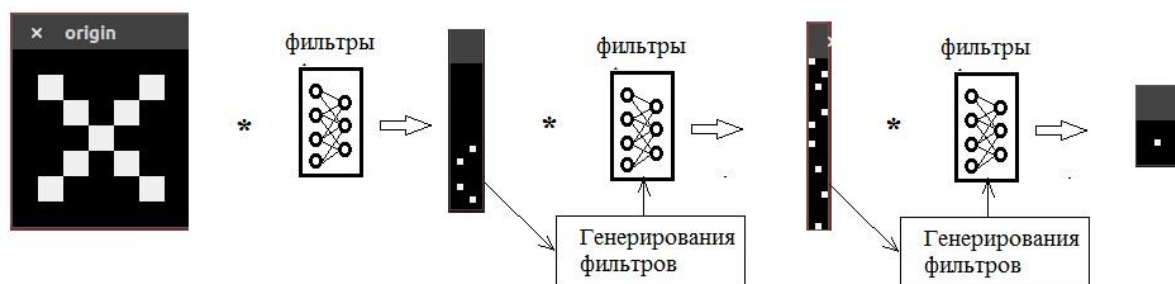


Рис. 2.6. Рекурсивное генерирование библиотеки фильтров-масок для одного образа [46].

В результате обучения одного нейрона (формировании одного образа), мы получили структуру подсети, выходной нейрон которой соответствует образу этого прообраза. При обучении другого нейрона (формирования другого образа), другая подсеть сгенерируется с другой структурой. Тогда сходство этих образов определяется количеством общих нейронов. Чем больше у них общих нейронов, тем более похожи эти образы. Кроме того, в конце процесса обучения, когда

нейрон для данного образа создан, протянем к нему «прямые связи» непосредственно от входного слоя (рис. 2.7). Такие «прямые связи» имеют два значения:

- они помогают быстро распознавать образы, хотя и с невысокой точностью;
- они помогают восстановлению исходных данных прообраза для передачи данных об этом образе другому роботу. Это один из видов языковой структуры.

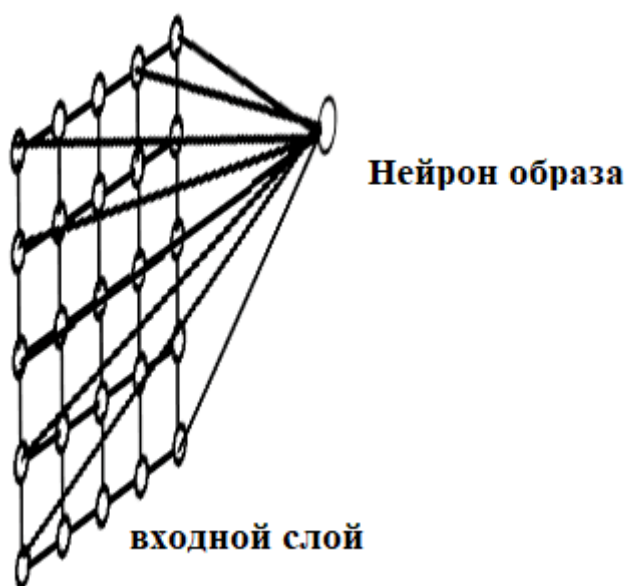


Рис. 2.7. Прямые связи нейрона одного образа с нейронами входного слоя.

Алгоритм обучения используется для каждого скалярного изображения, в результате обучения всех скалярных изображений получим массив подсетей для подсистемы ФРО. Полная схема сети для подсистемы ФРО изображена на Рис. 2.8.

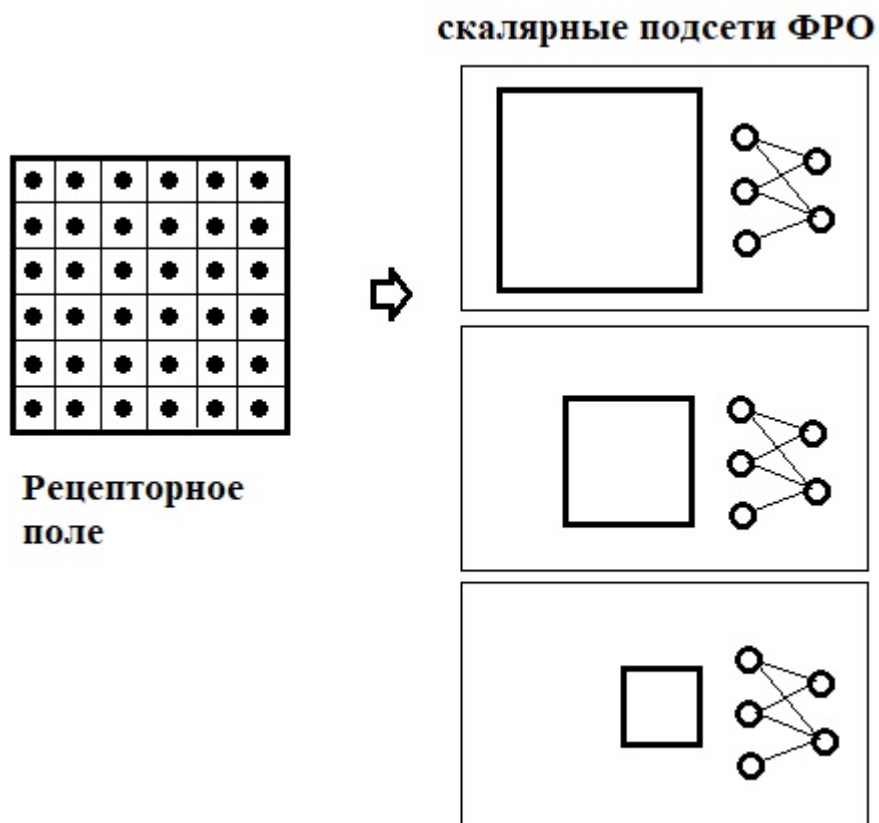


Рис. 2.8. Полная схема подсистемы ФРО.

Один из недостатков этого алгоритма – при обучении многих образов, количество новых нейронов будет большим, тогда возникают вычислительные проблемы большой размерности. Поэтому, мы делаем оптимальный процесс распознавания, который состоит из двух этапов:

- 1-й этап – быстро распознавать образ по прямым связям.
- 2-й этап – анализ деталей структуры распознанных образов в 1-ом этапе.

Полученные координаты прообразов совпадают с геометрическими центрами соответствующих им образов за счет описанной операции сжатия информации. Зная размер фильтров в каждом слое, можно восстановить исходные образы по условным координатам, что удобно использовать при подсчете распознанных образов на анализируемом изображении [46].

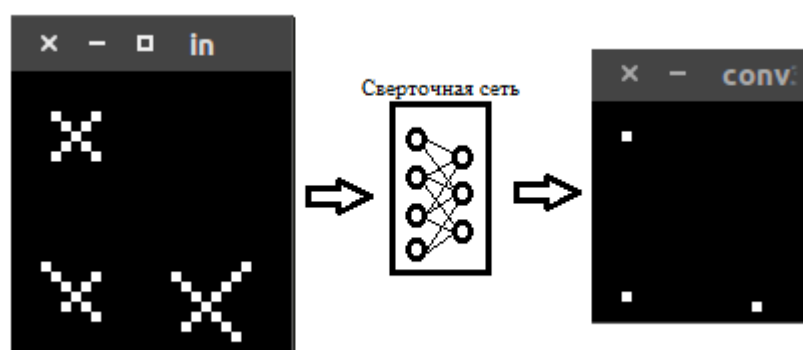


Рис. 2.9. Результат распознавания образа «X» [46].

Задача идентификации образов

При обучении нового образа в подсистеме ФРО сгенерируется структура с новыми нейронами, при этом структуры для разных образов отличаются друг от друга. Поэтому этот алгоритм можно использовать для решения задачи идентификации образов. Напомним, что существует только один выходной нейрон сгенерированной структуры при обучении одного образа. Это свойство нашей сети, по нашему убеждению, соответствует природным нервным системам. Ученые из университета Калифорнии в Лос-Анджелесе обнаружили в мозгу особые нейроны («нейроны места»), которые моментально кодируют ассоциации между людьми и местами, где они находятся. Еще пример - исследователи нашли в мозге в районе «отдела памяти» - гиппокампа одного испытуемого нейрон, который бурно реагировал только тогда, когда на глаза этому человеку попадалась фотография известной голливудской актрисы Дженнифер Энистон.. Появился даже официальный научный термин - «нейрон Дженнифер Энистон» [40]. Если на вход подсистемы ФРО поступают данные по прообразу этого образа в виде любой фотографии актрисы, написания или произнесения вслух ее имени, то выходной нейрон, отвечающий за образ Дженнифер Энистон в целом, срабатывает.. Это подтверждает, во-первых, положение теории метода ААУ, что при предъявлении прообраза сработает возможно несколько нейронов, распознающих отдельные части образа, но в этой пирамиде будет некий

конечный нейрон, отвечающий за конечный образ, и во-вторых, что в конечном образе ассоциированы прообразы разной физической природы – изображения, символные и звуковые идентификаторы.

2.2. Построение алгоритма самообучаемой нейроподобной сети для системы распознавания звуков

Обработка речевого сигнала

Смоделируем процесс распознавания речи, используя наши представления об основных физических и биологических механизмах этого процесса.

Как известно из физики, звуковые колебания характеризуются тремя атрибутами – амплитудой, частотой и фазой. Наша речь – это последовательность звуков различных частот. Поэтому, если рассматривать звук на коротких интервалах 5 – 25 мс, то характеристики звуковых колебаний можно считать постоянными [47].

На первом этапе обработки, звуковые колебания целесообразно нарезать на отрезки, называемые фреймами, с помощью движущегося временного окна с перекрытием (рис. 2.10).

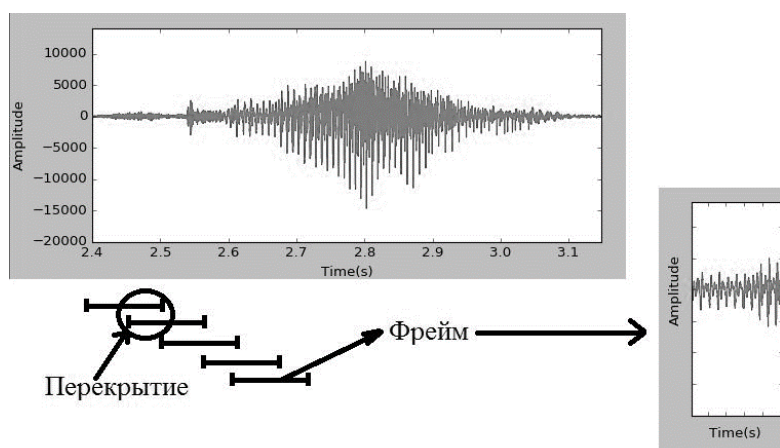


Рис. 2.10. Нарезка звуковых колебаний на фреймы с перекрытием.

Известно, что в слуховой улитке нашего органа слуха расположены рецепторные волосковые клетки, реагирующие на звуковые колебания воздушной

среды разной частоты. Каждая волосковая клетка реагирует на колебания своей частоты и, распознав таковое, генерирует нервный импульс, поступающий далее в нервную систему. Тем самым, можно сказать, что принцип распознавания звуков начинается с преобразования, подобного известному методу преобразования Фурье [34]. Информативными признаками, по которым ведется распознавание звуков, становятся мощности частотных составляющих регистрируемого спектра колебаний, а также временной ход вариаций мощности, характеризующий, в частности, относительные моменты времени появления и прекращения этих колебаний. Понятно также, что названные параметры носят в природе непрерывный характер, но все регистрирующие биологические устройства обладают, конечно, некоторыми пороговыми свойствами. Например, волосковая клетка регистрирует колебания, частота которых лежит в некотором диапазоне, а мощность превышает некоторый порог чувствительности. Существует также дискретизация по времени (периоды рефрактерности и т.п.). Соответственно, для получения этих признаков на каждом фрейме над сигналом следует провести дискретное преобразование Фурье [34]:

$$X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{N}kn},$$

где:

- N - количество значений сигнала, измеренных за период, а также количество компонент разложения;
- x_n , $n = 0, \dots, N-1$ - измеренные значения сигнала (в дискретных временных точках с номерами $n = 0, \dots, N-1$);
- X_k , $k = 0, \dots, N-1$ - комплексные амплитуды синусоидальных сигналов (мощности гармоник), составляющих исходный сигнал.

Упорядочиваем выборку спектров фреймов по времени соответствующих фреймов, в результате чего получаем спектрограмму (рис. 2.11). Примерно в

таким виде в мозг на распознавание поступают сигналы из биологического слухового аппарата. В науке и технике спектрограммы применяются для идентификации речи, анализа звуков, в радио, гидролокации, сейсмологии и в других областях.

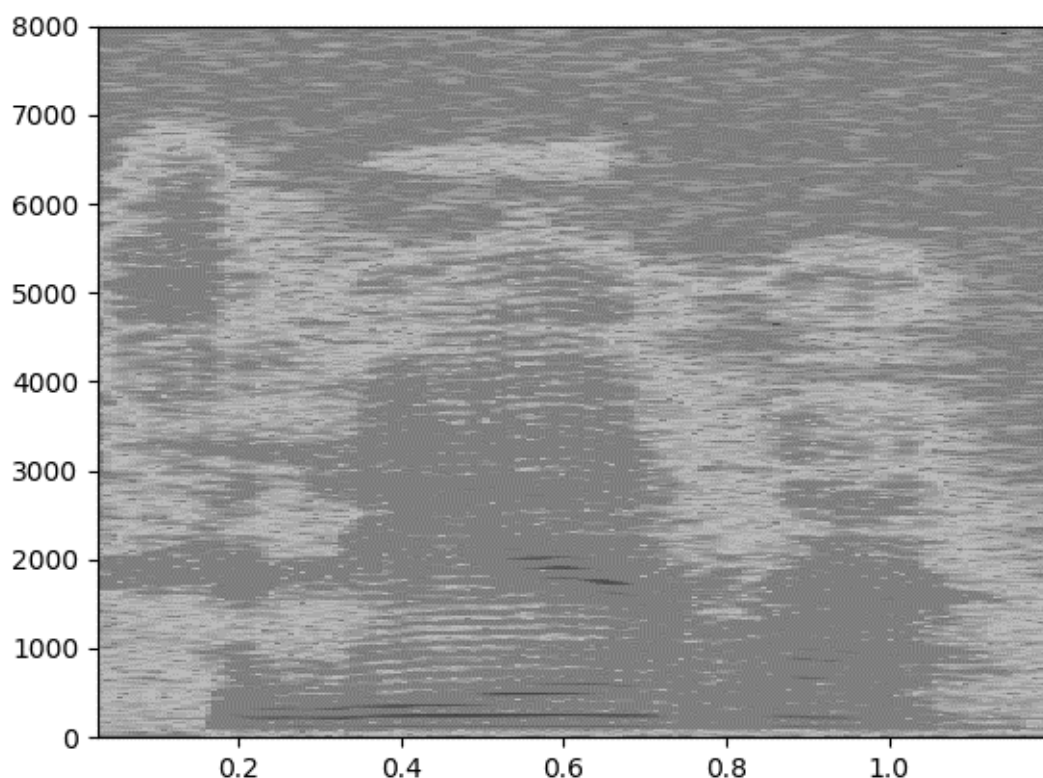


Рис. 2.11. Спектрограмма слова «слева».

Как сказано выше, если рассматривать звук на коротких интервалах 5 – 25 мс, то характеристики звуковых колебаний можно считать постоянными. Поэтому один фрейм можно использовать как входной сигнал от одного слога. Создаем блок датчиков, которые соответствуют разным частотам звуковых колебаний. Если амплитуда одной частоты превышает пороговое значение, то датчик этой частоты передает сигнал «1» дальше в нейронную сеть.

Рассмотрим вопрос дискретизации данных спектрограммы, так как нам надо привести сигналы к бинарному виду. Область слухового восприятия у нормально

слышащего человека ограничена по частоте и по силе звука. По частоте эта область охватывает диапазон от 20 до 22000 Гц (частотный диапазон слуха), а по силе от 20Дб до 130 Дб (динамический диапазон слуха).

Принято считать, что область речи, т.е. частотный и динамический диапазоны, необходимые для восприятия звуков речи, занимают лишь небольшую часть всей области слухового восприятия, а именно, по частоте от 500 до 8000 Гц и по силе от 50 до 90 Дб над порогом слышимости. Такое ограничение области речи по частоте и интенсивности может быть принято лишь весьма условно, так как оно оказывается действительным только в отношении наиболее важной для понимания речи области воспринимаемых звуков, но далеко не охватывает всех звуков, входящих в состав речи.

Целый ряд звуков русской речи, как, например, согласные [с], [з], [ц], содержит форманты, лежащие значительно выше 3000 Гц, а именно до 8000 Гц. Что касается динамического диапазона, то нужно учитывать, что уровень интенсивности тихого шепота равен 10-15 Дб, и что даже в громкой речи имеются такие составные элементы, интенсивность которых не превышает уровня обычной шепотной речи, т. е. 25 Дб; к числу таких элементов относятся, например, некоторые глухие согласные. Следовательно, для полноценного различения на слух всех звуков речи необходима сохранность всей или почти всей области слухового восприятия, как в отношении частоты, так и в отношении интенсивности звука [4].

Применяем модель «банка фильтров» - это модель частотной чувствительности человеческого уха, в которой используются «Мел-кепстральные коэффициенты» [5,47]. Идея метода заключается в том, чтобы разделить интересующий диапазон частот на полосы и измерять общую энергетику в каждой полосе. Фильтры могут перекрываться по частотам (рис. 2.12).

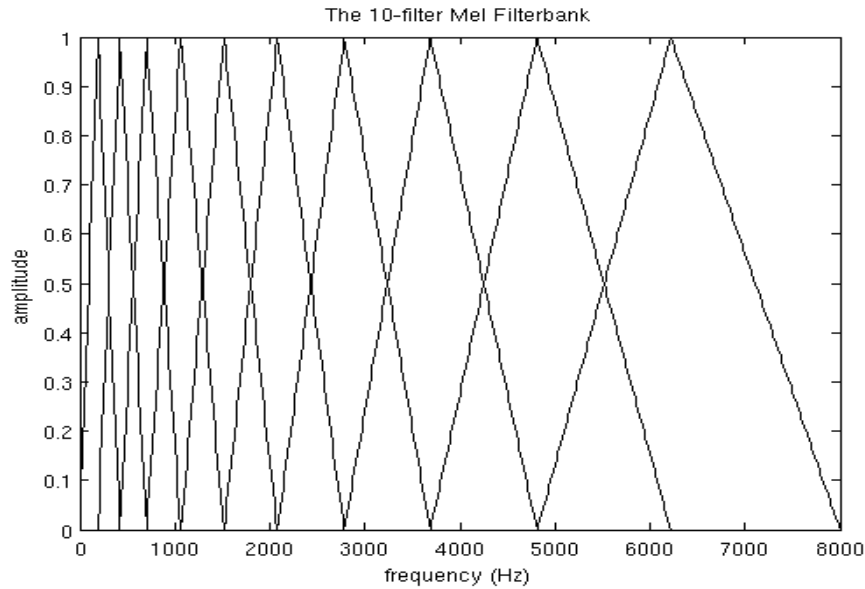


Рис. 2.12. Мел-кепстральные коэффициенты.

Обозначим через $s(t)$ исходный речевой сигнал. Тогда сигнал на выходе из фильтра представляет собой краткосрочное спектральное представление исходного сигнала в момент времени t . Очевидно, что в этой модели каждый фильтр обрабатывает речевой сигнал независимо. Они вычисляются следующим образом. При равномерном расположении центральная частота f_i i -го фильтра определяется как

$$f_i = \frac{F}{N}, 1 \leq i \leq N,$$

где F - частота дискретизации, а N – количество фильтров, необходимых для покрытия всего диапазона частот.

Треугольные фильтры, на которые мы будем "накладывать" спектр нашего сигнала:

$$H_m(k) = \begin{cases} 0 & k < f[m-1] \\ \frac{(k-f[m-1])}{(f[m]-f[m-1])} & f[m-1] \leq k < f[m] \\ \frac{(f[m+1]-k)}{(f[m+1]-f[m])} & f[m] \leq k \leq f[m+1] \\ 0 & k > f[m+1] \end{cases}$$

где:

$$f[m] = \left(\frac{N}{F_s}\right)B^{-1}\left(B(f_1) + m\frac{B(f_h)-B(f_1)}{M+1}\right),$$

где M - количество фильтров (сколько коэффициентов мы хотим получить).

$$B(f) = 1125 \ln\left(1 + \frac{f}{700}\right)$$

Расположение в соответствии с особенностями человеческого восприятия.

Логарифмированная энергия сигнала для каждого фильтра:

$$S[m] = \ln\left(\sum_{k=0}^{N-1} \text{abs}(X[k])^2 H_m[k]\right), 0 \leq m < M.$$

В результате получается набор сигналов $S(m)$, $1 \leq m \leq M$, которые представляют оценку энергии речевого сигнала в каждой полосе. Признаки, получаемые из этой модели, преобразуются в бинарный вид с помощью пороговой функции

$$S_i = \begin{cases} 1, S_m \geq \rho \\ 0 \end{cases},$$

где S_m - признак вектора модели банка фильтров речевого сигнала в каждой спектральной полосе, ρ - порог для этой полосы (рис. 2.13).

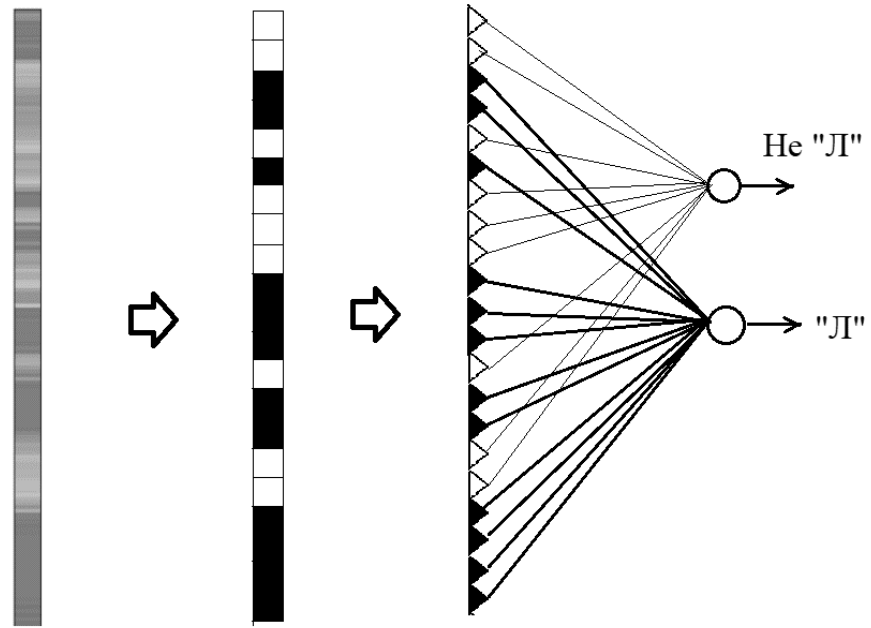


Рис. 2.13. Пример вектора буквы «Л» в бинарном виде [47].

Чтобы уменьшать вероятность ложной тревоги, используем «нейросетевой фильтр» [46,47], который служит для фильтрации набора значимых сигналов и для исключения набора незначимых.

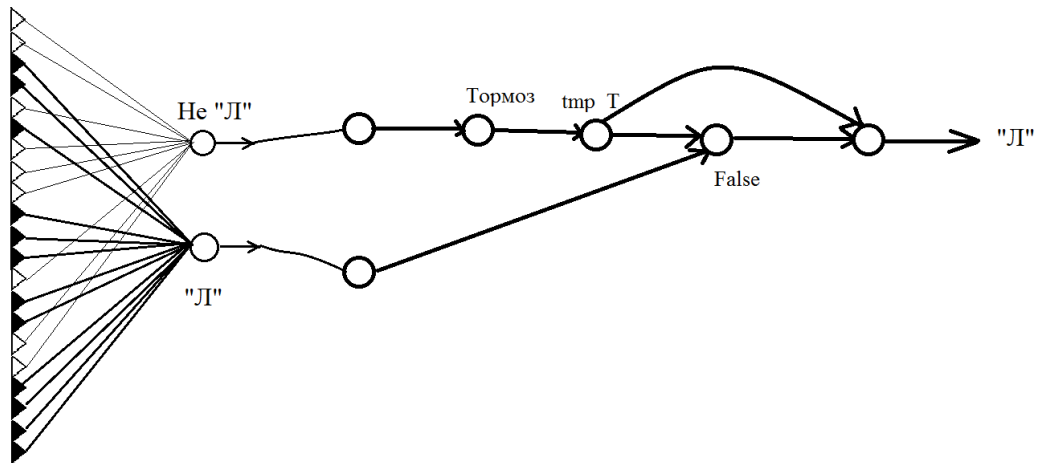


Рис. 2.14. Пример автоматически сгенерированной структуры нейроноподобной подсети для определения вектора буквы «Л» [47].

Структуры для распознавания всех букв строятся аналогично тому, как показано выше для буквы «Л». При поступлении полных звуковых колебаний

одного слова, мы получаем последовательность распознанных букв. Например, слово «слева» поступает на вход сети так:

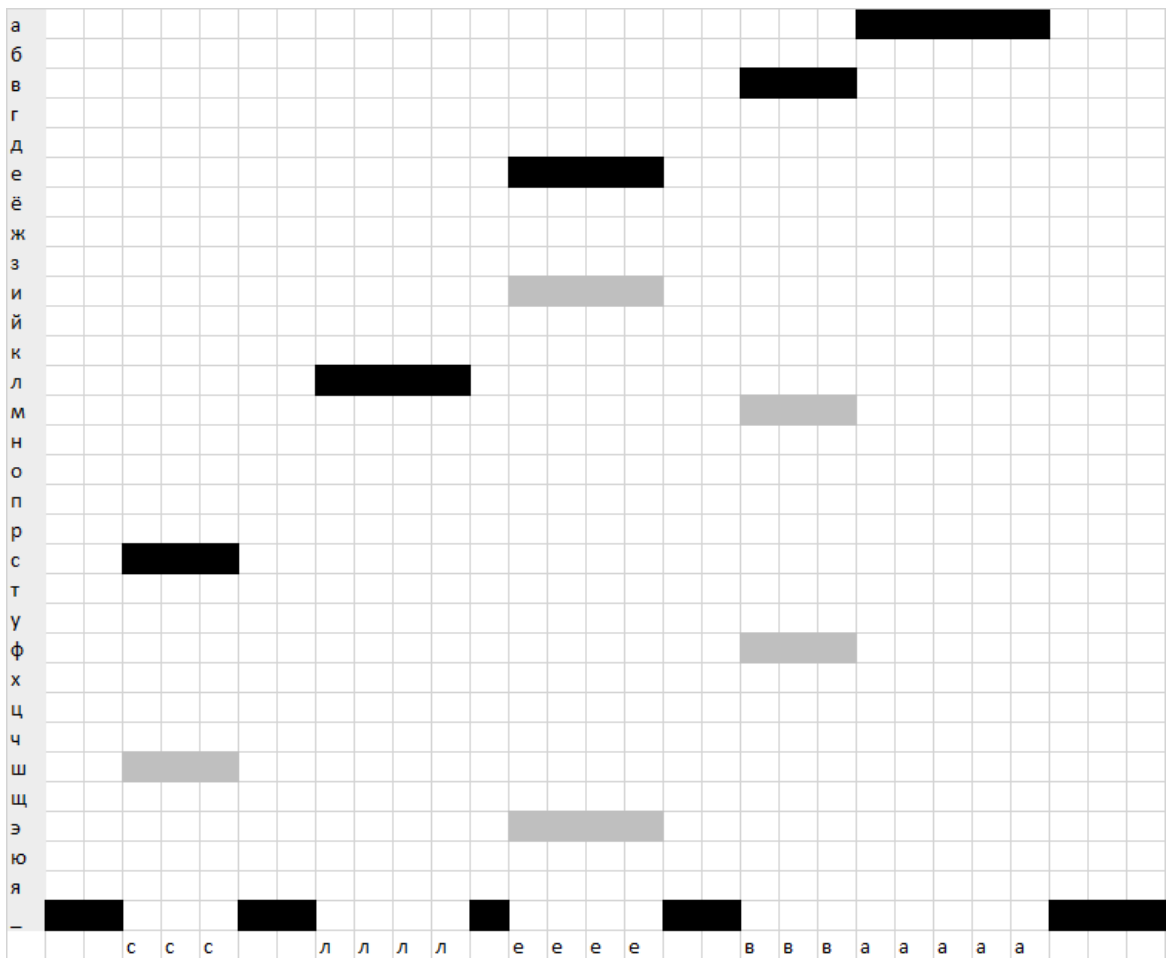


Рис. 2.15. Результаты распознавания простых звуков.

В результате предыдущего этапа, мы получили последовательность букв. В данном примере речевой прообраз образа слова «слева» состоит из 5-х основных элементов:

«с → с → с → л → л → л → л → е → е → е → е → в → в → в → а → а → а»

Или в сокращенном виде:

«с → л → е → в → а».

Для распознавания последовательных образов мы используем два метода:

- нейрон 2-го типа [5] (рис. 2) ;
- структуру очереди на основе нейрона 1-го типа (описано в 1-ой главе).

«Структура очереди» на основе нейронов 2-го типа (рис. 2.16) [47].

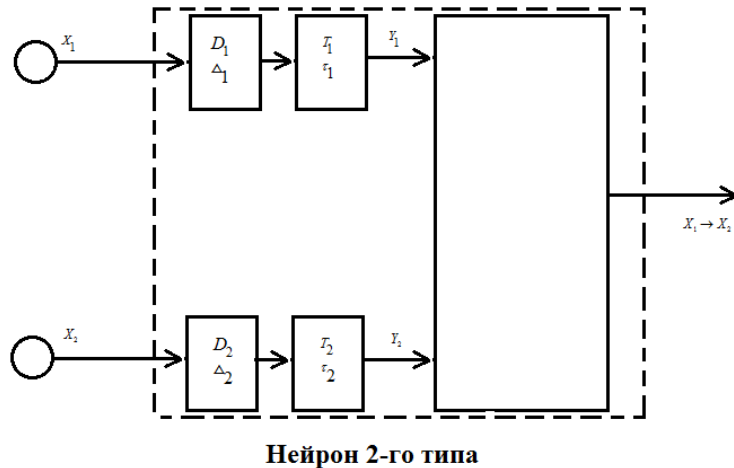


Рис. 2.16. Если $\tau_2=1$, а $\tau_1=2$, то нейрон 2-го типа будет срабатывать только тогда, когда сигналы x_1 и x_2 последовательно поступают на вход нейрона, в других случаях нейрон не будет срабатывать [47].

Как сказано выше, слово «слева» состоит из 5 основных элементов « $c \rightarrow л \rightarrow e \rightarrow в \rightarrow a$ ». В качестве фильтров для каждого звука буквы, мы можем распознавать букву, которая соответствует поступившему звуку. Для создания подсети для конкретного слово «слева» (последовательности звуков) производится процесс автоматической генерации всех возможностей «структур очереди» последовательности двух звуков (рис. 2.17).

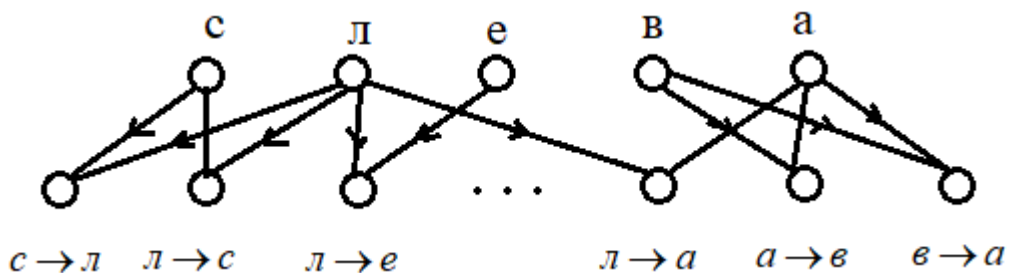


Рис. 2.17. Генерация всех возможностей «структур очереди» последовательности.

При поступлении последовательности звуков слова «слева» происходит процесс обучения нейронов подсети. Через некоторое время, когда число повторения поступающих звуков достаточно большое, то некоторые нейроны из сгенерированных нейронов становятся обученными. Обученные нейроны будут сохранены, а необученные нейроны удаляются.

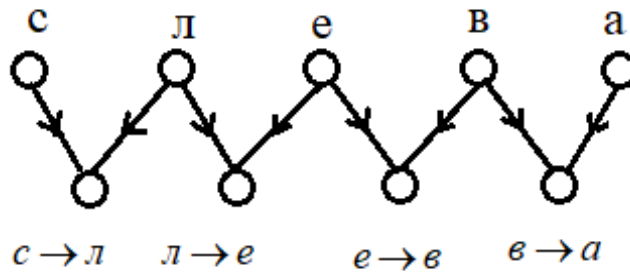


Рис. 2.18. Результат обучения слова «слева».

Процессы генерации и обучения так же повторяются для следующего слоя, который состоит из обученных сгенерированных нейронов. Эти процессы генерации и обучения для одного слова заканчиваются, когда число сгенерированных нейронов становится равно 2 (рис. 2.20). Тогда только один из двух сгенерированных нейронов будет обучен – это нейрон образ слова.

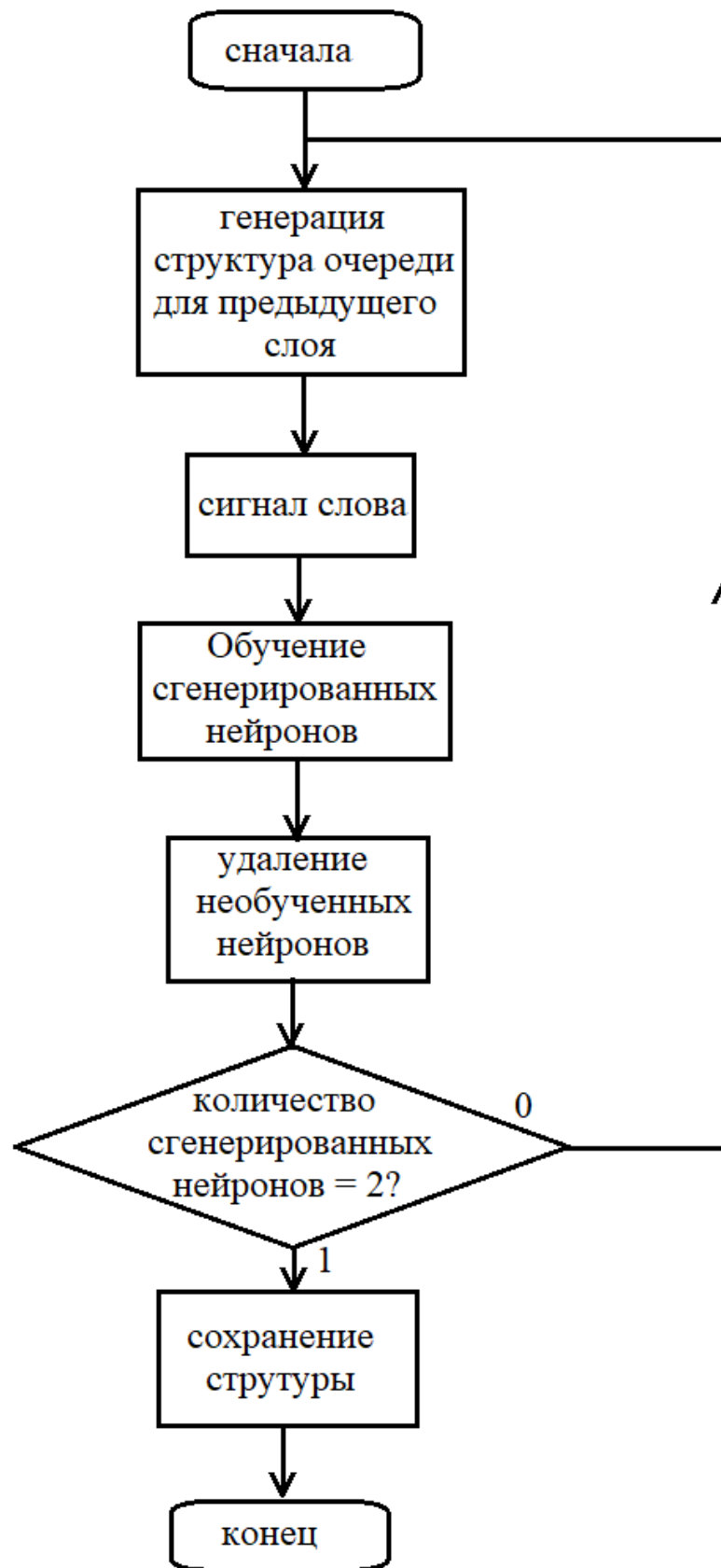


Рис. 2.20. Алгоритм генерации подсети для одной последовательности звуков.

Тогда, подсеть слова «слева» будет такой:

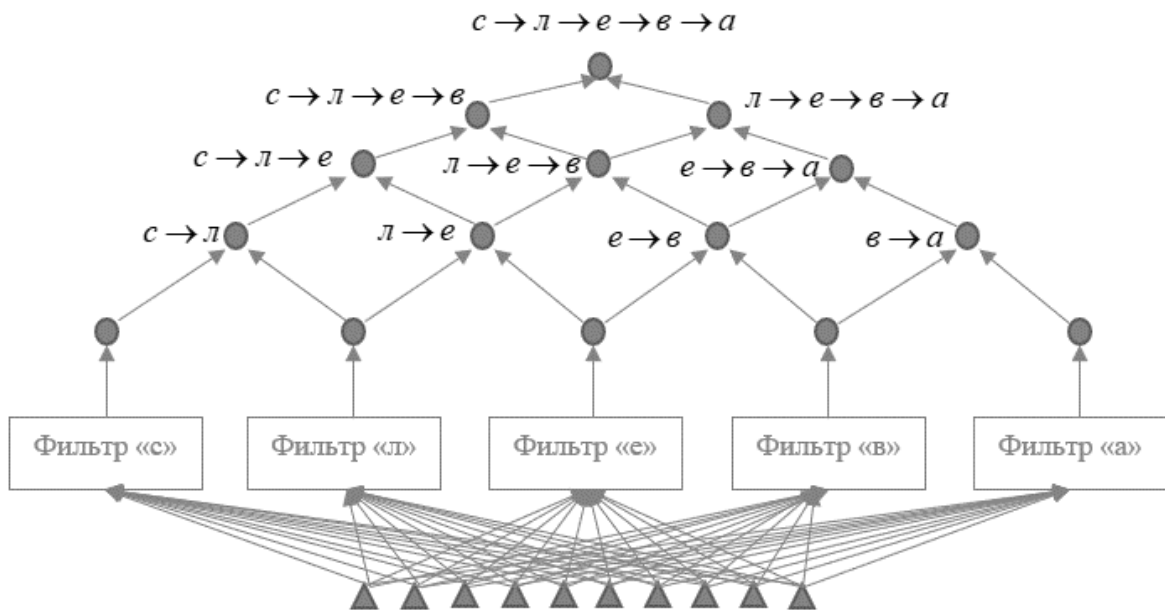


Рис. 2.21. Полная структура сети для распознавания речевого образа «слева» [47].

С помощью двух элементарных структур - «структуры фильтра» и «структуры очереди» на основе нейрона 2-го типа, мы можем создать нейроподобную сеть для распознавания любых речевых образов [47]. Процессом обучения является генерация эталонного образа, либо обучение одной из уже имеющихся нейронных структур, либо требуется формирование такой структуры в онтогенезе, если таковой структуры у организма не оказалось. Эти процессы вполне могут быть автоматизированы с помощью разработанных нами алгоритмов генерации сети нейронов [46,47,49].

2.3. Выводы по главе 2.

Таким образом, мы смоделировали и представили здесь алгоритмы, реализующие наши биоморфные представления о входном тракте языкового процесса, возникающего и реализующегося в сети нейронов, соответствующих концептуальной модели нервной системы [1]. Мы показали, как посредством

датчиков, регистрирующих мощности определенных частот в слышимом звуке, и откликающихся на превышение их порогов чувствительности бинарными сигналами, информация о языковых единицах (словах) преобразуется в последовательность бинарных векторов, которые подаются на вход распознающей сети из описанных в [5] нейронов, и приводят к процессу обучения нейронов (описано в главе 1/1.1/процесс обучения нейрона), а затем к распознаванию этих специфических последовательностей, что интерпретируется, как распознавание слов. Как отмечалось во Введении, распознавание слов является только началом языкового процесса, который затем продолжается посредством установления ассоциативных связей образов языковых идентификаторов с образами реальных объектов, явлений или процессов. В результате в системе возникает возможность посредством произнесения языковых идентификаторов (слов) вызывать в нервной системе (и в системе ААУ) распознавание образов реальных явлений и заставлять ее реагировать на них определенным способом. Представленные здесь результаты, развивающие работы авторов из группы, в которой выполнялась настоящая работа [1,5,37-39], могут быть использованы в двух направлениях: при построении моделей, отражающих механизмы работы биологических систем управления, а также для построения технических систем распознавания речи [47], обладающих свойствами самообучения.

ГЛАВА 3. Реализация основных программных модулей адаптивной программной системы со свойствами языкового интерфейса на примере системы автономного адаптивного управления

Программы систем управления метода ААУ могут быть реализованы на различных языках программирования. Все эти системы построены по одной схеме, изображенной на Рис. 1 Введения. В зависимости от конкретных задач управления, модули реализуют на основе нейроподобных сетей или просто на основе структур данных, таких как матрицы, списки, кортежи, двоичное дерево и т.д. С помощью современного компьютера и языков программирования, взаимодействия между основными модулями метода ААУ могут быть реализованы в парадигме как последовательного, так и параллельного программирования.

Для программирования системы управления, описанной в 1-ой главе, мы выбрали язык программирования Python, так как данный язык содержит самые современные механизмы повторного использования кода и удобочитаемый синтаксис. Python также является высокоуровневым объектно-ориентированным языком программирования (ООП). В нем мы можем создать классы, которые очень удобны для создания нейронов разных типов метода ААУ. Особенно Python позволяет достаточно просто и эффективно использовать процессы и разделяемую память для параллельного программирования в стиле множества потоков и общей памяти.

3.1. Реализация ядра системы управления ААУ

Ядро-ААУ [49] является главным модулем, который позволяет проектировать и создавать многопроцессные реализации нейроподобных систем. Нейроны разных типов были реализованы как классы на объектно-

ориентированном языке программирования - Python. Тогда нейрон Жданова как адаптивный нейроноподобный элемент системы работает как отдельный процесс. В этом классе синапсы нейрона легко создаются списками (list) в Python. Совокупность списка нейронов и их связей между собой создает нейроноподобную сеть. Класс нейрона Жданова реализован следующим способом:

```
class Neuron:
    def __init__(self, ID):
        self.ID = ID # идентификатор нейрона
        self.out_neurons = [] # выходные связи
        self.in_neurons = [] # входные связи
        self.ADAPTED = 0 # 0 - Нейрон необучен, 1 - нейрон обучен
        self.OUT = 0 # Выходной сигнал: 1 - распознан, 0- не распознан
        self.TYPE_OF_NEURON = "" # тип нейрона 1,2,3
```

С помощью этого абстрактного класса нейрона, мы можем создать нейронную сеть со своей структурой. Каждый нейрон будет экземпляром этого класса, а связи с предыдущими и с следующими слоями будут описаны в списках `self.out_neurons` и `self.in_neurons`.

Например:

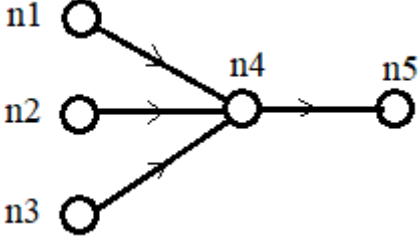
КОД	Нейронная сеть
<pre># создать нейроны n1 = NEURON (ID="n1") n2 = NEURON (ID="n2") n3 = NEURON (ID="n3") n4 = NEURON (ID="n4") n5 = NEURON (ID="n5") # создать связи n4.in_neurons [n1, n2, n3] n4.out_neurons [n5]</pre>	

Рис. 3.1. Пример структуры нейронной сети.

Чтобы сохранить структуру нейроноподобных сетей, мы использовали формат JSON или простой формат текста. Структурный формат JSON

поддерживает массивы и вложенность собственных структур, которые удобны для сохранения структуры нейроноподобных сетей. Пример на рисунке 3.2.

1) Простой формат текста	2) Формат JSON для сохранения структуры сети
<pre>n1:n4 n2:n4 n3:n4 n4:n5</pre>	<pre>{ "neurons": [{n1,n2,n3,n4,n5}] "synapses": [{ "ID_IN": "n1", "ID_OUT": "n4" }, { "ID_IN": "n2", "ID_OUT": "n4" }, { "ID_IN": "n3", "ID_OUT": "n4" }, { "ID_IN": "n4", "ID_OUT": "n5" }] }</pre>

Рис. 3.2. Форматы файла сохранения структуры сети.

Нейроны Жданова в своей наиболее простой форме работают только с бинарными сигналами, а также их выходной сигнал равен либо 0, либо 1. Тогда, модули системы управления ААУ, которые реализованы на основе нейроноподобных сетей, работают с бинарными потоками.

3.2. Протокол обмена данными между модулями системы управления ААУ

В теории метода ААУ входные и выходные данные для каждого модуля являются бинарными потоками. Согласование по времени различных нейроноподобных сетей можно осуществлять разными методами, чтобы

обеспечивать синхронизацию работы нейронов в системе и взаимодействие между сетями. В этой реализации, мы использовали буферную схему (структурный FIFO стек), которая будет пропускать на выход подсистемы ФРО только поток бинарных сигналов. Стек работает следующим способом.

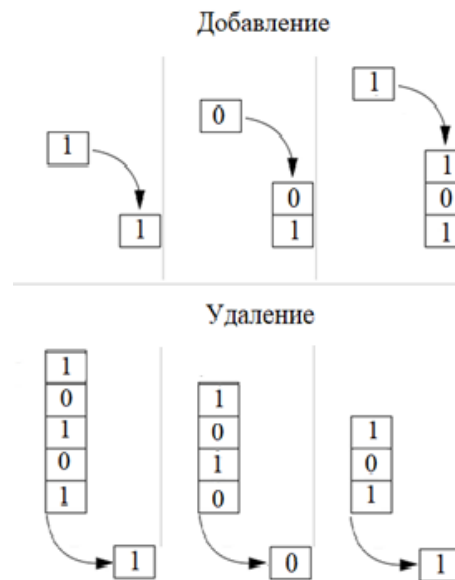


Рис. 3.4. Принцип работы структурных данных «стека».

Класс стека был реализован следующим:

```
class Stack:
    def __init__(self):
        self.stack = []
    def pop(self):
        if self.is_empty():
            return None
        else:
            return self.stack.pop()
    def push(self, val):
        return self.stack.append(val)
    def peak(self):
        if self.is_empty():
            return None
        else:
            return self.stack[-1]
    def size(self):
        return len(self.stack)
    def is_empty(self):
        return self.size() == 0
```

С помощью этого протокола, каждый модуль может работать в отдельном процессе, а результат работы каждого модуля записывается в общую память. Это

позволяет нам проектировать систему управления на основе архитектуры параллельных вычислительных систем (рис. 3.4).

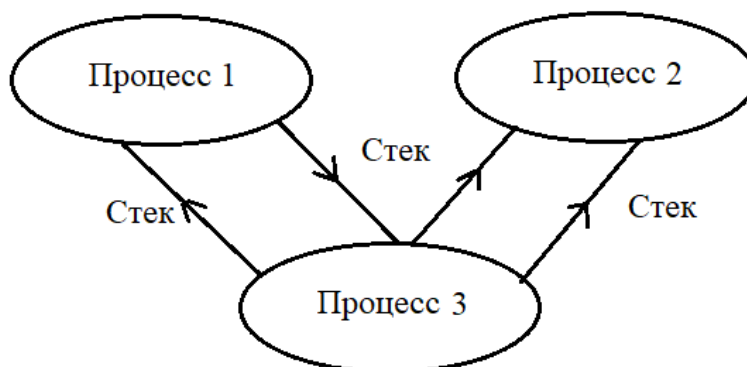


Рис. 3.4. Протокол обмена данными.

3.3. Реализация модулей модификации структур нейроноподобных сети

Матрица событий

В матрице событий [49] элементы E_{ik} накапливают статистику совпадений срабатываний пары нейронов (i,k) . Поэтому матрицу событий удобно использовать для построения ассоциативного слоя (ассоциативная память). В момент времени t поступают векторы распознанных образов из разных подсистем ФРО, если i -й элемент 1-го вектора и k -й элемент 2-го вектора равны 1, то элемент E_{ik} увеличивается на 1.

В языке программирования Python, библиотека Numpy предназначена для создания многомерных массивов элементов с большим набором математических функций и для операций с этими массивами. Матрица событий так же легко создается с помощью этой библиотеки.

Программный код создания матрицы:

```
import numpy as np
MatrixEvent = np.zeros((m,n), dtype=np.uint8)
```

Где m – размер 1-го вектора, n – размер 2-го вектора.

При поступлении векторов распознанных образов, выполняется функция обновления элементов матрицы событий:

```
def update(self, v1, v2):
    N = len(v1)
    M = len(v2)

    for i in range(N):
        for k in range(M):
            # Если v1[i]=1 и v2[k]=1
            if (v1[i] + v2[k]) == 2:
                if self.MatrixEvent[i,k]>=0:
                    self.MatrixEvent[i,k] += 1
```

Если размеры векторов изменяются при появлении новых образов в подсистеме ФРО, то размерность матрицы легко расширяется с помощью операции:

```
if Old_N < New_N:
    pad_width = New_N-Old_N
    self.MatrixEvent = np.lib.pad(self.MatrixEvent,
    [(0,pad_width),(0,0)], 'constant', constant_values=0)
```

Каждый раз, когда выполняется операция обновления элементов матрицы событий, проводится проверка значений всех элементов матрицы. Если значение какого-либо элемента E_{ik} превышает пороговое, то блок матрицы событий будет посылать команду подсистеме ФРО создать новый нейрон, входными нейронами которого будут нейроны с номерами i и k .

Дерево событий

Дерево событий [47,49] (ДС) используется для накопления статистики совпадений срабатываний нескольких активных нейронов, чтобы находить закономерные комбинации образов для построения системы ФРО. Узел этого дерева описывается следующим классом:

```
class Node:
    def __init__(self):
        self.l = None # левая ветка
        self.r = None # правая ветка
        self.learned = False #
        self.count = 0 # число совпадения
```

Сначала дерево события будет иметь ссылку на объект «Node», который является корнем дерева:

```
class Tree:
    def __init__(self):
        self.root = Node()
        self.size = 0
```

При поступлении вектора распознанных образов $O(t) = (1, 0, 0, 1, 1, 0, \dots, 0, 1)$, проводится процесс построения дерева. Для этого рассмотрим по очереди каждый элемент вектора. Если рассматриваемый элемент вектора $O(t)$ равен «1», то ДС строит правый узел относительно предыдущего или увеличивает счетчик правого узла, если он уже создан. Если рассматриваемый элемент вектора $O(t)$ равен «0», то ДС строит левый узел относительно предыдущего или увеличивает счетчик левого узла, если он уже создан.

```
def _add(self, vector, node):
    if(len(vector)>0):
        if(vector[0] == 0):
            # создать левую ветку
            if(node.l != None):
                self._add(vector[1:], node.l)
            else:
                node.l = Node()
                self._add(vector[1:], node.l)
        else:
            # создать правую ветку
            if(node.r != None):
                self._add(vector[1:], node.r)
            else:
                node.r = Node()
                self._add(vector[1:], node.r)
```

В каждый такт времени ведется учет совпадений сигналов от всех нейронов выходного слоя подсистемы ФРО (рис. 3.5).

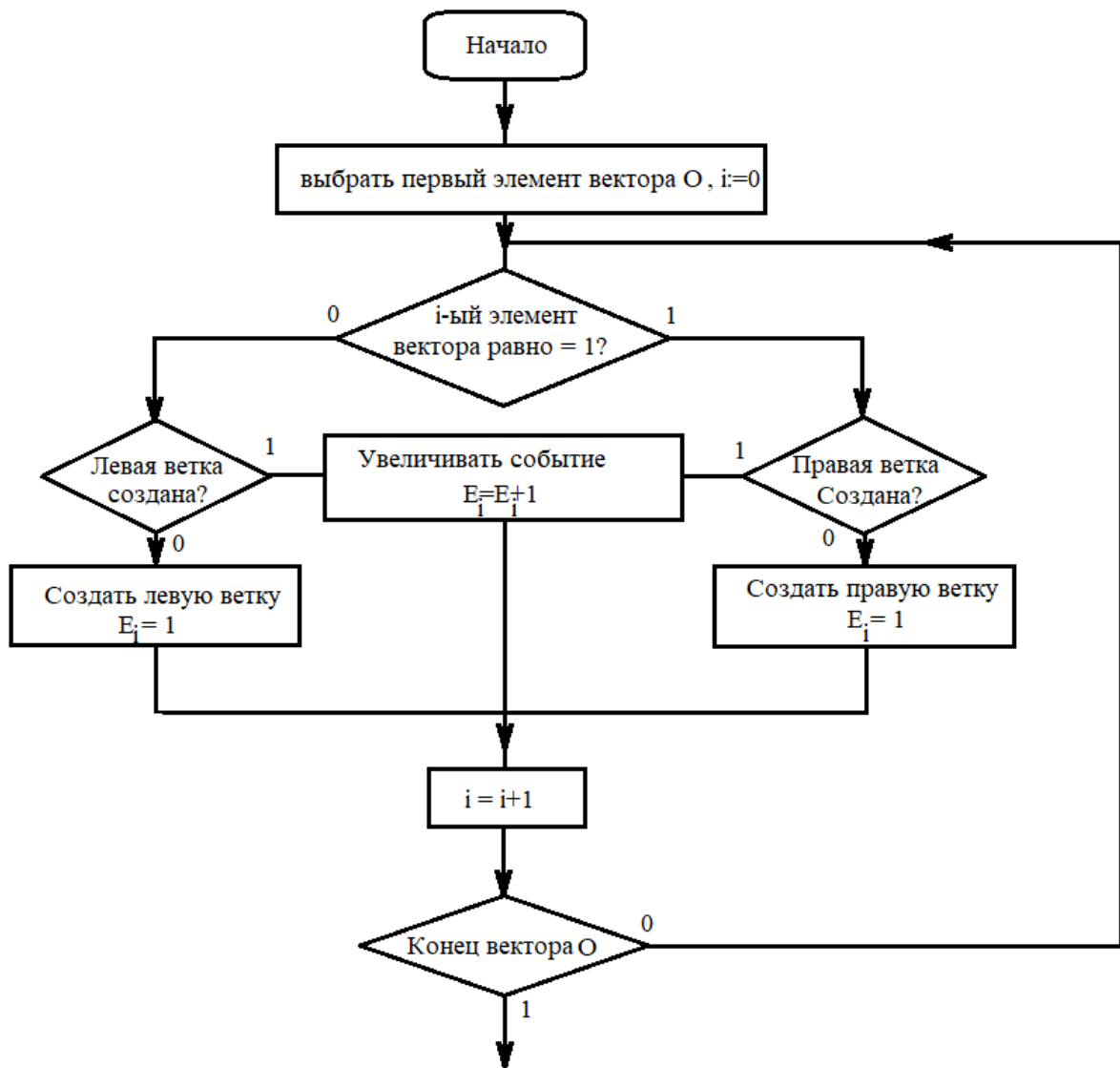


Рис. 3.5. Алгоритм построения дерева событий.

В определенный момент времени для некоторого узла ДС происходит событие, когда значение этого узла $E > L$, и возникает кандидат на рождение нового нейрона от нейронов из этого узла до корня [49]. При этом новый нейрон (образ) добавляется в выходной слой. Выходной вектор сети - это вектор состояний распознанных образов. При росте нейронной сети размер выходного вектора тоже изменяется.

3.4. Реализация аппарата эмоций метода ААУ

EMOTION – «аппарат эмоций». «Аппарат эмоций» - это системообразующая подсистема в методе ААУ, задающая целевые функции и качественные критерии для принятия решений управляющей системой [5]. Как сказано в **1-й главе**, подсистема ФРО содержит в себе базовые образы (набор датчиков), у каждого из которых есть своя априорная качественная («эмоциональная») оценка, которую записали в аппарате эмоций. Таблица «образ-оценка» является априорной информацией УС в системе ААУ. Если в подсистеме ФРО появился новый образ, то его качественная оценка определяется по следующей формуле:

$$E_{obj} = \sum_{i=0}^m (E_i \times x_i) , \quad (3.2),$$

где:

- E_{obj} - эмоциональная оценка нового образа;
- $\{ E_i \}$ - эмоциональные оценки базовых образов;
- $X = (x_1, x_2, x_3, \dots, x_{m-1}, x_m)$ – карта нового образа.

Программный код аппарата эмоций:

```
global BASE_OBJECTS_SCORE = []

def Emotion(base_objects_score, map_of_object):
    ##### ПАРАМЕТРЫ #####
    # base_objects_score - вектор базовых оценок
    # map_of_object - вектор входных сигналов образа
    #####
    # Векторное умножение
    score = base_objects_score * map_of_object
    return score
```

3.5. Выводы по главе 3

В этой главе, мы представили алгоритмы и язык программирования реализации системы управления метода ААУ. Предложены методы и протоколы обмена данными между модулями системы управления. Эти модули являются основными инструментами для проектирования системы управления ААУ и языкового человеко-машинного интерфейса.

ГЛАВА 4. Прикладная система автономного адаптивного управления с языковым человеко-машинным интерфейсом для БПЛА

4.1. Модель виртуальной среды для моделирования полета БПЛА на UNITY3D

Для демонстрации новых методов реализации системы ААУ с голосовым интерфейсом, мы в качестве объектов управления использовали модели виртуальных БПЛА (коптеров) (рис. 4.1).

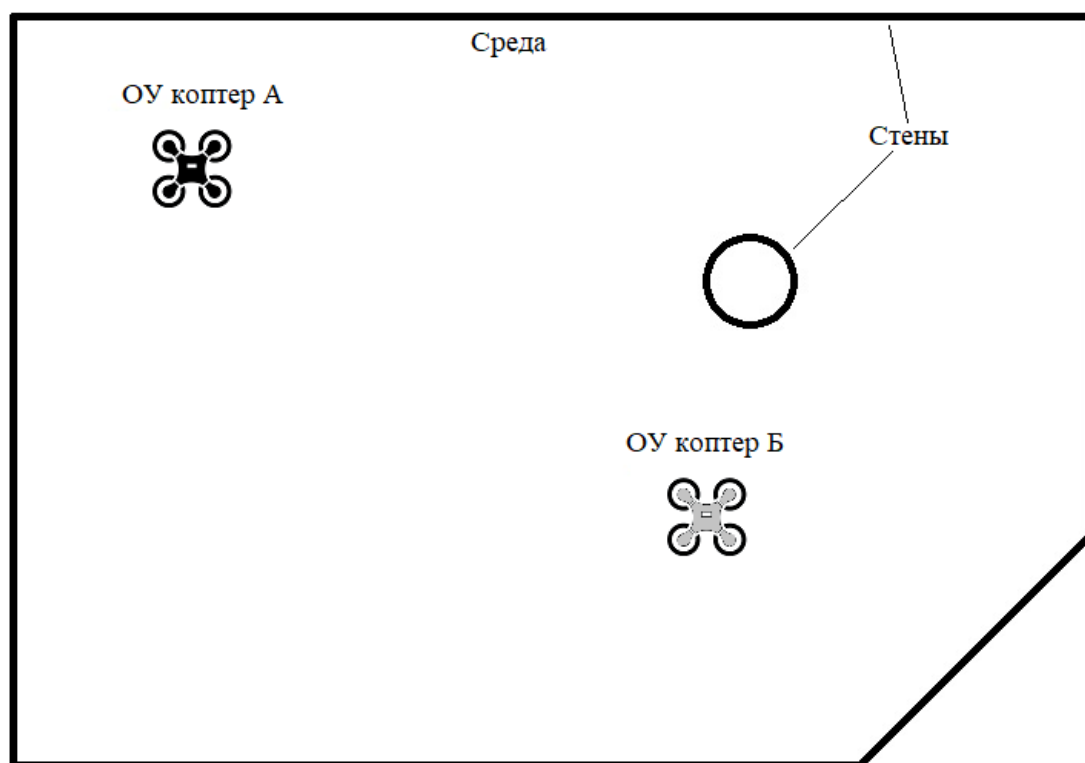


Рис. 4.1. Виртуальная модель БПЛА в среде UNITY3D.

Реализованные модели БПЛА были способны выполнять следующие действия:

- изменять угол рыскания (направления полета) на θ градусов;
- двигаться вперед или назад со скоростью C ;

где угол θ и скорость C являются свободными параметрами этой виртуальной среды. В этом примере мы выбрали $\theta \in \{-60, -55, \dots, -5, 0, 5, 10, \dots, 60\}$ градусов и $C \in \{0, 20, 40, \dots, 100\}$ см/с. Действия данных БПЛА образуются из комбинаций значений (θ, C) . Количество действий БПЛА зависит от сложности действий, которые ОУ БПЛА может совершать.

Пример набора движений БПЛА показан на рисунке 4.2. Коптер может двигаться период, назад, поворот влево (вправо) и оставаться на месте.

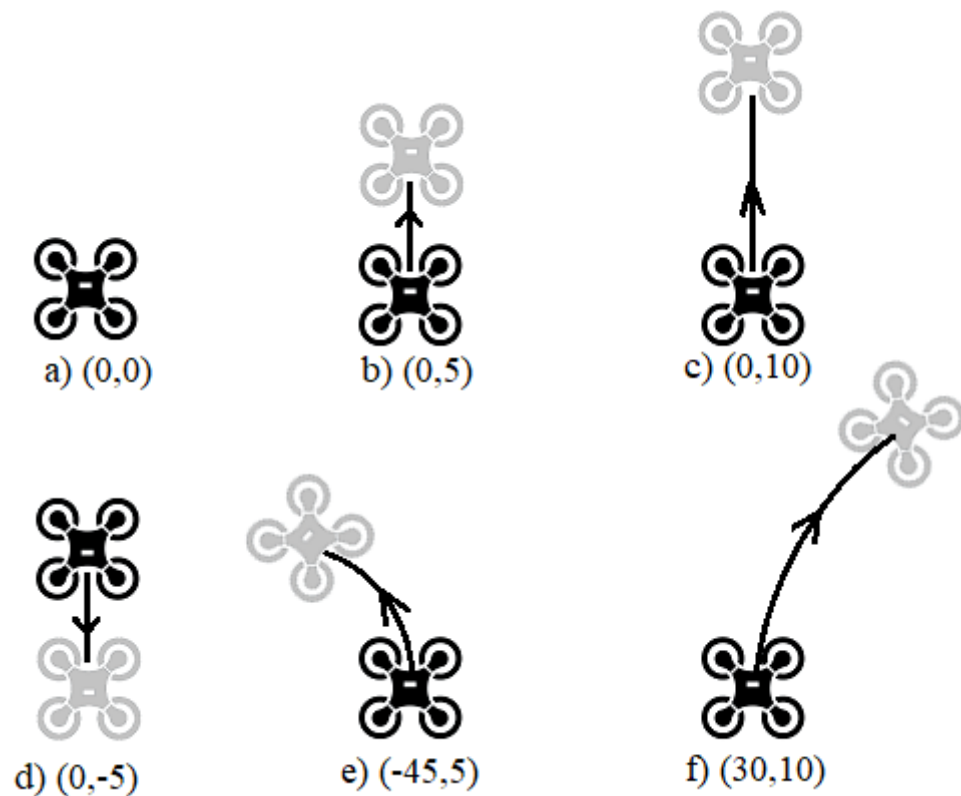


Рис. 4.2. Примеры действий (движений) БПЛА.

На БПЛА установили сенсоры, которые могут определять локальные координаты преследуемого БПЛА. Сенсоры БПЛА а состоят из 2-х датчиков:

- 1-й датчик определяет расстояние до преследуемого БПЛА. В простом варианте, мы не измеряем точное расстояние, а датчик

возвращает условные расстояния («ближе», «не очень далеко», «далеко»);

- 2-й датчик определяет угол в локальной системе координат, под которым виден преследуемый БПЛА.

С двумя датчиками мы можем определить, в каком секторе обзора находится преследуемый БПЛА (рис. 4.3).

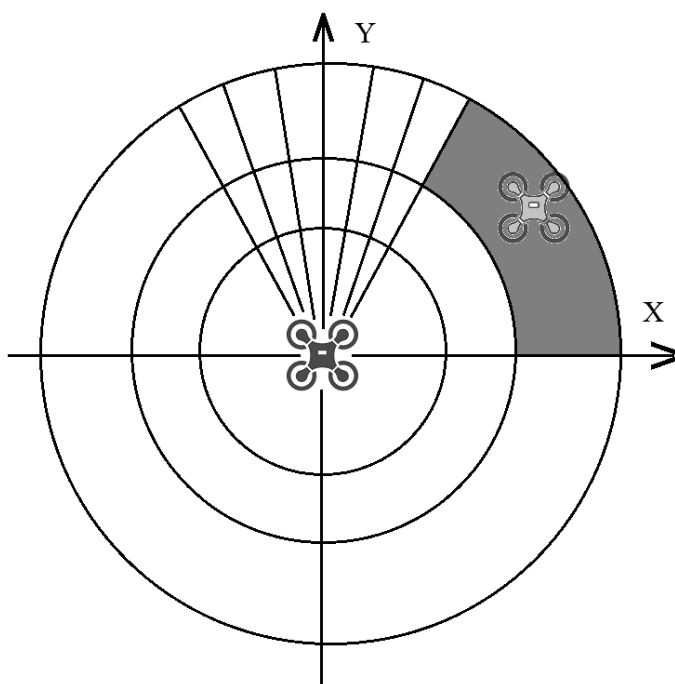


Рис. 4.3. Пример определения БПЛА положения преследуемого БПЛА в локальной системе координат.

В виртуальном помещении, где мы установили два БПЛА, первый БПЛА является убегающим, преследуемым, а второй БПЛА догоняет его и только этот БПЛА имеет адаптивную систему управления, поэтому ниже будем называть его «адаптивным БПЛА».

Убегающий БПЛА может двигаться разными способами:

- нет движения – БПЛА покоится;

- случайный выбор – случайным способом выбирается угол и выставляется максимальная скорость. БПЛА движется в этом направлении вплоть до определения препятствия;
- по маршруту - определяется очередная точка движения, и БПЛА поворачивается в её направлении, выставляется максимальная скорость. После достижения точки, определяется следующая точка и процесс повторяется.

Адаптивный БПЛА получает команды от своей бортовой системы ААУ, которая реализована на Python, и выполняет эти команды для заданной цели.

4.2. Постановка задачи управления БПЛА с голосовым интерфейсом

1-я задача

Адаптивный БПЛА самообучается двигаться так, чтобы догнать убегающий БПЛА и держать его в центральном секторе обзора (рис. 4.4). Сначала адаптивный БПЛА двигается в случайных направлениях, и его адаптивная система управления начинает накапливать знания о взаимодействиях со средой. Алгоритмом задано, что если убегающий БПЛА находится далеко и не в центральном секторе обзора, то адаптивный БПЛА получает отрицательную эмоцию, а в противных случаях он получает положительную эмоцию.

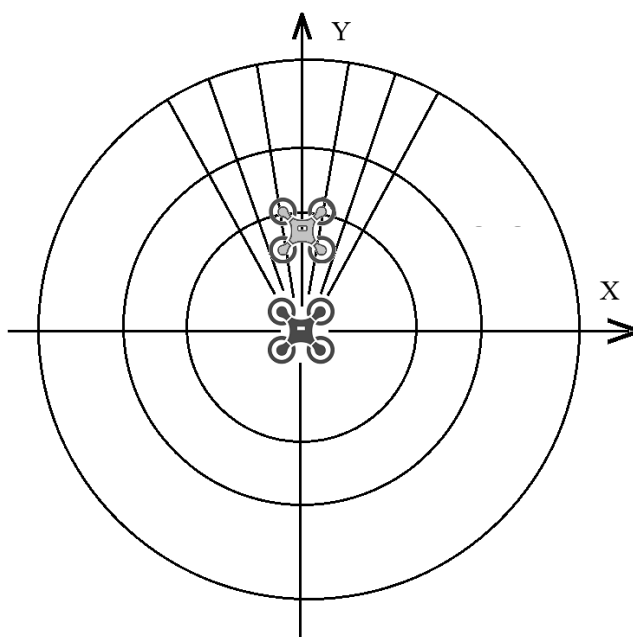


Рис. 4.4. Целевое состояние для системы управления догоняющего адаптивного БПЛА.

2-я задача

Мы добавили в УС БПЛА подсистему распознавания звуков, которая к этому времени уже обучилась распознавать два слова «слева» и «справа». Теперь в УС БПЛА работают одновременно две подсистемы ФРО:

- система распознавания реальных ситуаций;
- система распознавания звуков.

Распознанные образы обеих подсистем ФРО будут участвовать в процессе обучения УС. Система управления БПЛА продемонстрировала алгоритм генерации нейроноподобных конструкций, порождающих язык. Мы стремились к тому, чтобы УС управляла БПЛА как с помощью распознавания визуальной информации, так и с помощью распознавания звуковой информации. Результат этого эксперимента показывает эффективную идентификацию образов и успех формирования нейроноподобных конструкций, порождающих язык на основе ААУ.

4.3. Алгоритм функционирования системы управления БПЛА

Для проведения экспериментов с разработанной системой управления и исследования ее свойств нами была разработана имитационная система (рис. 4.5.).

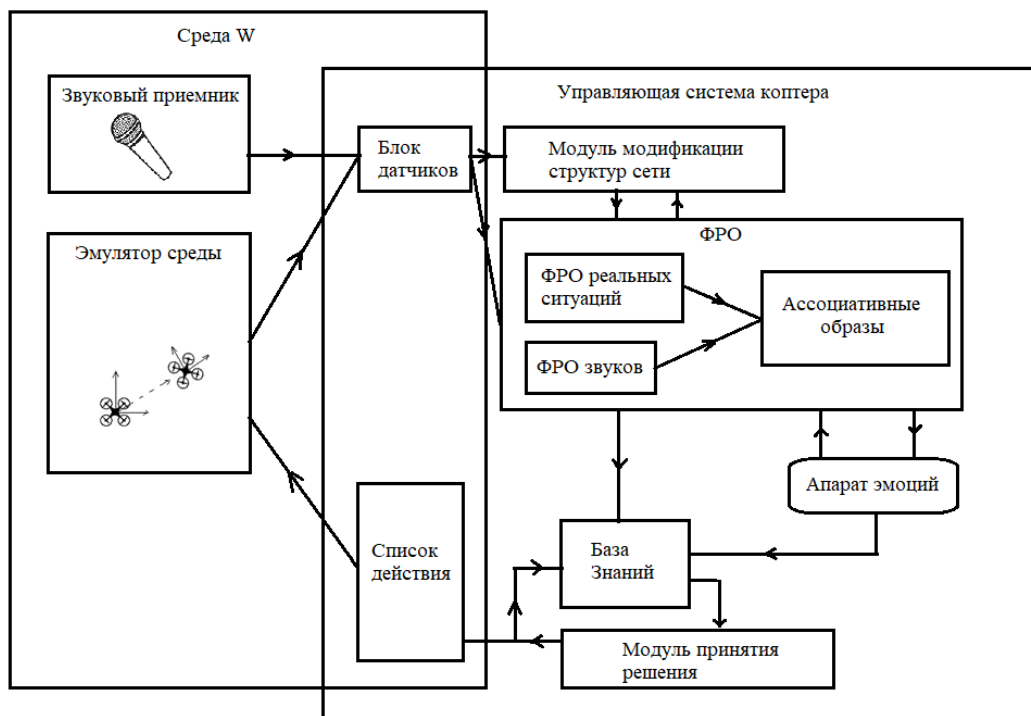


Рис. 4.5. Схема имитационной системы для исследования свойств управления БПЛА на основе метода ААУ.

Имитационная система была реализована в среде программного пакета **Python**. Система управления состоит из следующих элементов:

- Рупеух – это модуль, который является ядром системы управления;
- SimpleWebsocket – модуль, обеспечивающий обмен данными с виртуальной средой;
- VISUAL – модуль для изображения структур нейроподобных сетей;
- BZ – база знаний;
- CONVERTER – модуль преобразования входных данных в бинарный вид.
- EMOTION – аппарат эмоций;

- FROEYE – система распознавания реальных ситуаций;
- FROEAR – система распознавания звуков, как голосовой интерфейс;
- TreeEvent – Модуль анализа совпадения срабатываний нескольких нейронов;
- MatrixEvent – модуль анализа совпадения срабатывания двух нейронов;
- ListAction – список возможных действий БПЛА.

CONVERTER – модуль, который преобразует входные сигналы из виртуальной среды в бинарный вид, потому что подсистема ФРО в данном случае работает только с бинарными векторами. Векторы после обработки будут поступать на входы подсистемы ФРО.

Кодирование вектора положения – Функция «pos2bin».

Сообщение от модели виртуальной среды БПЛА записано в виде JSON:

```
{
  Name: «adaptive copter», // Название коптера
  angle: «30», // угловая скорость
  velocity: «10», // линейная скорость
  distance_to_wall: «20», // расстояние до стены
  distance_to_other_copter: «30», // расстояние до бегающего коптера
  angle_to_other_copter: "45" // угол к бегающему коптеру
}
```

Рис. 4.6 формат данных, получаемых из виртуальной среды.

Функция «pos2bin» использует только два входных параметра `distance_to_other_copter` и `angle_to_other_copter`, чтобы кодировать их в бинарный вектор. Для кодирования расстояния до преследуемого БПЛА мы использовали три диапазона расстояний ([0,5], (5,20],[20;...]). Бинарный вектор, полученный в результате кодирования, состоит из 3-х элементов. Значение элемента вектора будет равно 1, если расстояние находится в своем диапазоне, а в противном случае будет равно 0. Например, так, как на рисунке 4.7.

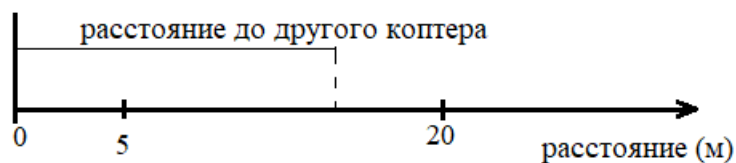


Рис. 4.7. Пример вектора кодирования расстояние $v_l = [0,1,0]$ - «не далеко».

Аналогично, для кодирования направления, образы направлений мы разделили на 6 частей, которые изображены на рис. 4.8. Тогда вектор кодирования направления состоит из 6-ти элементов. Значение элемента вектора будет равно 1, если направление находится в этом секторе, а в противном случае оно будет равно 0.

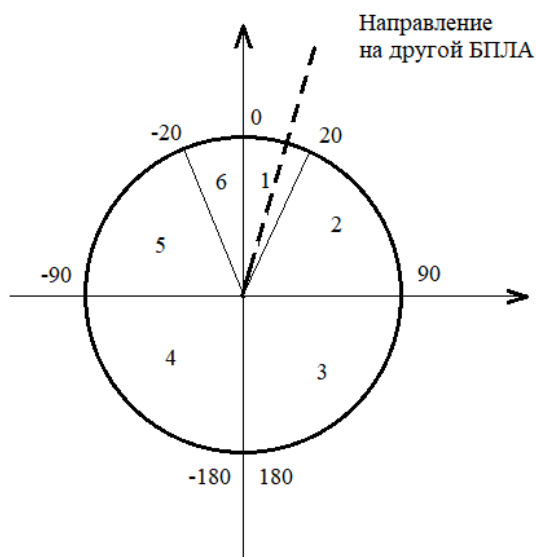


Рис. 4.8. Пример случая, в котором вектор кодирования будет равен

$$v_\theta = [1,0,0,0,0,0].$$

Векторы кодирования расстояния и направления образуют входной вектор, который дальше поступает на подсистему ФРО положения (FROEYE).

Кодирование звуков для голосового интерфейса

Алгоритм предобработки речевых сигналов был описан в главе 2. Все модули предобработки реализованы на языке Python. Звуковые сигналы принимаются с помощью стандартной библиотеки «pyaudio», которая является инструментом для работы со звуком. В результате предобработки в модуле «MFCC» мы получаем вектор признака «Мел-кепстральные коэффициенты». Признаки, получаемые из этой модели, преобразуются в бинарный вид с помощью пороговой функции:

$$S_i = \begin{cases} 1, S_m \geq \rho \\ 0 \end{cases}, \quad (4.1),$$

где S_m - признак вектора модели банка фильтров речевого сигнала в каждой спектральной полосе, ρ - порог для этой полосы.

В результате мы получаем вектор кодирования звуков размерности 24x1, который будет поступать на подсистему ФРО звуков (FROEAR) для распознавания. Все этапы изображены в картине рис.4.9.

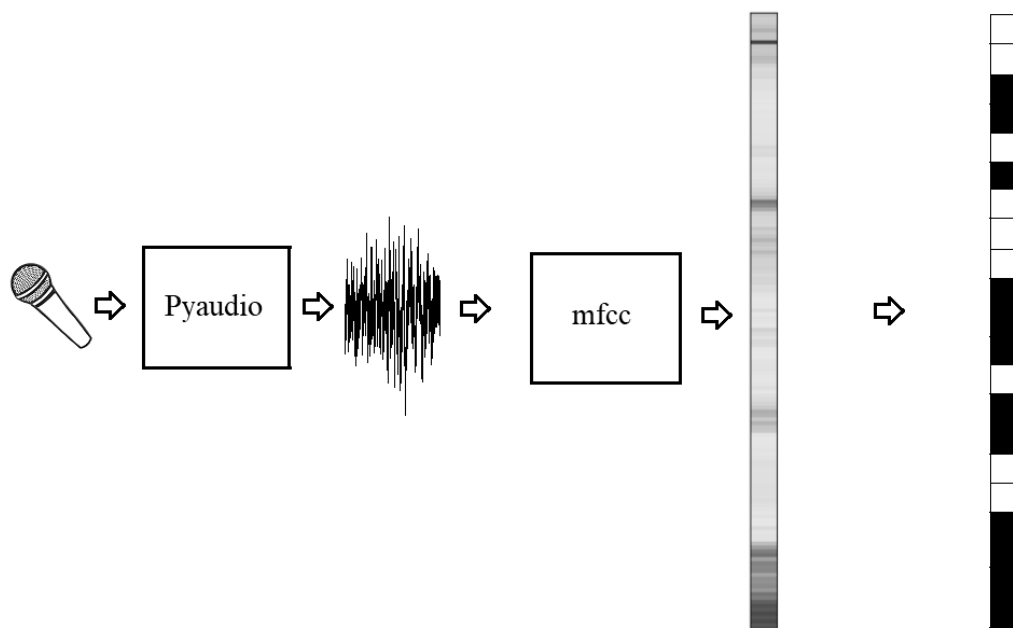


Рис.4.9. Схема этапов кодирования звуков.

FROEAR – система распознавания звуков. Была реализована для распознавания двух слов «слева» и «справа». Входной вектор, полученный из модуля *Кодирование звуков* будет поступать на входы этого модуля. Структура сети сохраняется в файле «net.json», модуль ядра «**pyneurox**» может загрузить этот файл структуры и создать нейронную сеть. Выходной вектор этой сети состоит из двух элементов, соответствующих словам («слева», «справа»). Принцип реализации описан выше в гл.2.

FROEYE – подсистема распознавания текущих ситуаций. Как сказано выше, входной вектор образуется из вектора кодирования расстояния $v_r = [0,1,0]$ и вектора кодирования направления $v_\theta = [1,0,0,0,0,0]$. То есть, входной вектор имеет размерность 9×1 . С помощью модуля ядра «**pyneurox**», была создана нейронная сеть с 9 рецепторами. Сначала сеть пустая, но при получении потока входных векторов, дерево событий «**TreeEvent**» накапливает данные по входным векторам, и сеть начинает достраиваться (рис. 4.10).

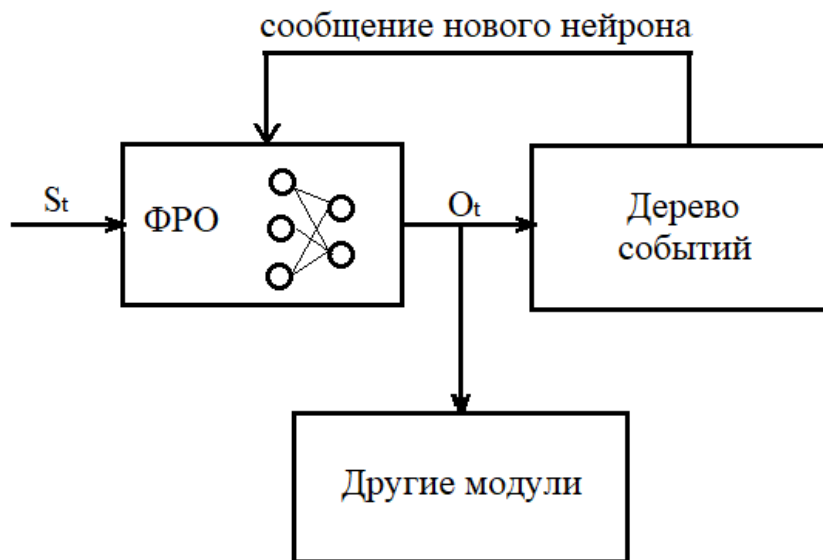


Рис. 4.10. Алгоритм построения нейроноподобных сетей.

В процессе работы все данные от виртуальной среды и речевые сигналы поступают в модуль «CONVERTER». На выходе модуля «CONVERTER» формируются два вектора кодирования. Вектор кодирования положения убегающего БПЛА V_{Pos} поступает на модуль «FROEYE» (подсистема ФРО реальных ситуаций). А вектор кодирования звуков V_{sound} поступает на «FROEAR» (подсистема ФРО распознавания звуков). На выходах этих подсистем ФРО формируются распознанные образы, образующие векторы $O_{Pos}(t)$ и $O_{Sound}(t)$.

В одном направлении вектор образов $S(t-1)$ - названный *вектором ситуации*, который образуется векторами $O_{Pos}(t)$ и $O_{Sound}(t)$, поступает в базу знаний. База знаний включает режим обучения для тех нейронов, которые получили на вход сигнал «1». Далее, блок принятия решения выбирает все обученные действия и вычисляет оценки для каждого действия по следующей формуле:

$$S_j = \sum_{i=1}^m o_i w_{ji} , \quad (4.3),$$

где S_j - эмоциональная оценка результата при совершении действия A_j , o_i - эмоциональная оценка образа O_i , w_{ji} - значение i -го веса j -го нейрона БЗ. Эмоциональная оценка образа O_i определяется блоком аппарата эмоции «EMOTION».

Вычисляется процент обучения базы знаний для этой ситуации:

$$P = (A - A_{not}) / A, \quad (4.4)$$

где - A_{not} количество необученных действий, A - количество всех возможных действий.

Если все оценки в множестве оценок $\{S_j\}$ низкие, то блок принятия решений случайно выбирает либо действие из множества необученных действий A_{not} , либо действие из множества обученных действий с вероятностью P . Выбранное действие $A(t)$ отправляется в виртуальную среду.

В следующий момент времени вектор ситуации $O(t+1)$, который является результатом события $S(t-1) \& A(t)$, поступает на входы базы знаний. Все нейроны в режиме обучения обучаются событию $(S(t-1) \& A(t)) \rightarrow O(t+1)$. После обучения нейроны базы знаний переключаются в режим работы. И так далее, полный контур управления и обучения повторяется.

В другом направлении, $O_{Pos}(t)$ поступает на входы дерева событий ДС, чтобы найти закономерности совпадения срабатывания нейронов подсистемы ФРО текущих ситуаций. Если возникает кандидат на рождение нового нейрона, то новая структура отправляется в подсистему ФРО для модификации структуры нейронной сети. После своего рождения новый нейрон будет участвовать в контуре управления системы. При этом наращивается и нейроноподобная база знаний.

Кроме того, векторы $O_{Pos}(t)$ и $O_{Sound}(t)$ поступают на входы матрицы событий. Элементы E_{ik} матрицы событий накапливает статистику совпадения срабатываний пары нейронов (i,k) . В определенный момент времени для некоторой пары (i,k) , происходит событие, когда значение $E_{ik} > L$, и возникает кандидат на рождение нового нейрона в подсистеме ФРО от нейронов с номерами i и k . Если нейроны i и k находятся в разных подсистемах ФРО, то новый нейрон с конструкцией идентификатора (рис. 4.11) добавится в подсистему ФРО.

«Новорожденный» нейрон имеет пороговое значение меньше 0,5, чтобы он срабатывал в случаях, если сработал только один из входящих нейронов.

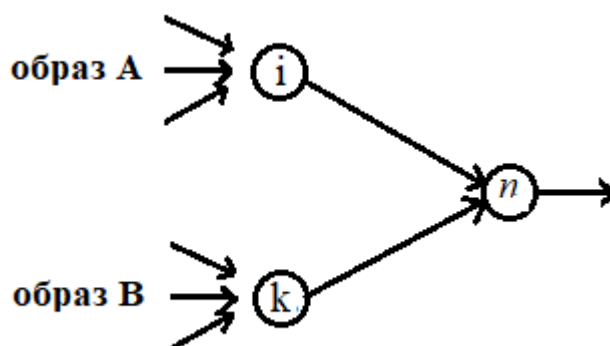


Рис. 4.11. Конструкция ассоциативного обучения.

Согласно теории ААУ, биологический мозг должен реализовывать целевую функцию выживания. В технической прикладной системе «выживание объекта» можно обусловить выполнением им некоторой полезной для человека заданной целевой функции. Эта функция зависит от вектора ситуаций, который сформирован подсистемой ФРО. Например, необходимо увеличивать время задержки убегающего БПЛА в центральном секторе обзора догоняющего БПЛА и минимизировать расстояние до убегающего БПЛА за определенной промежуток времени. Качество управления определяется средним значением оценок за N итераций:

$$F_{\text{cost}} = \frac{\sum_{i=0}^N \theta_i + \sum_{i=0}^N l_i}{N}, \quad (4.5),$$

где:

- N – такты времени;
- θ_i – оценка по направлению в момент i ;
- l_i – оценка по расстоянию в момент i .

Функционирование системы управления, реализованной на основе метода ААУ, описано на рис. 4.12.

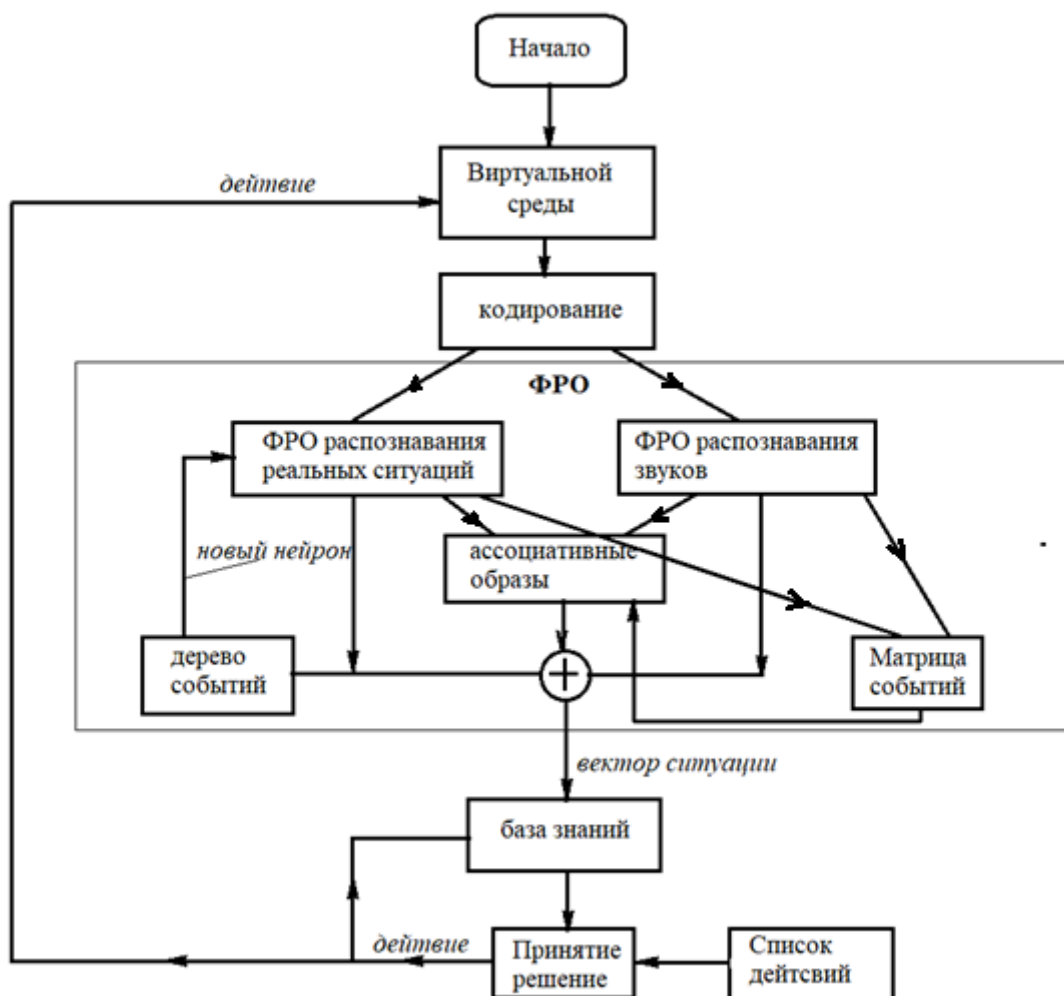


Рис. 4.12. Полный контур управления и алгоритм построения ассоциативной памяти.

4.4. Результаты экспериментов

Виртуальная среда

Виртуальная среда была реализована инженерами Научно-внедренческой компании «Интеллект». Интерфейс этой программы изображен в рис. 4.13. Из виртуальной среды, мы получаем данные состояний обоих БПЛА, а также отправляем команды управления БПЛА через модуль «Диспетчер».

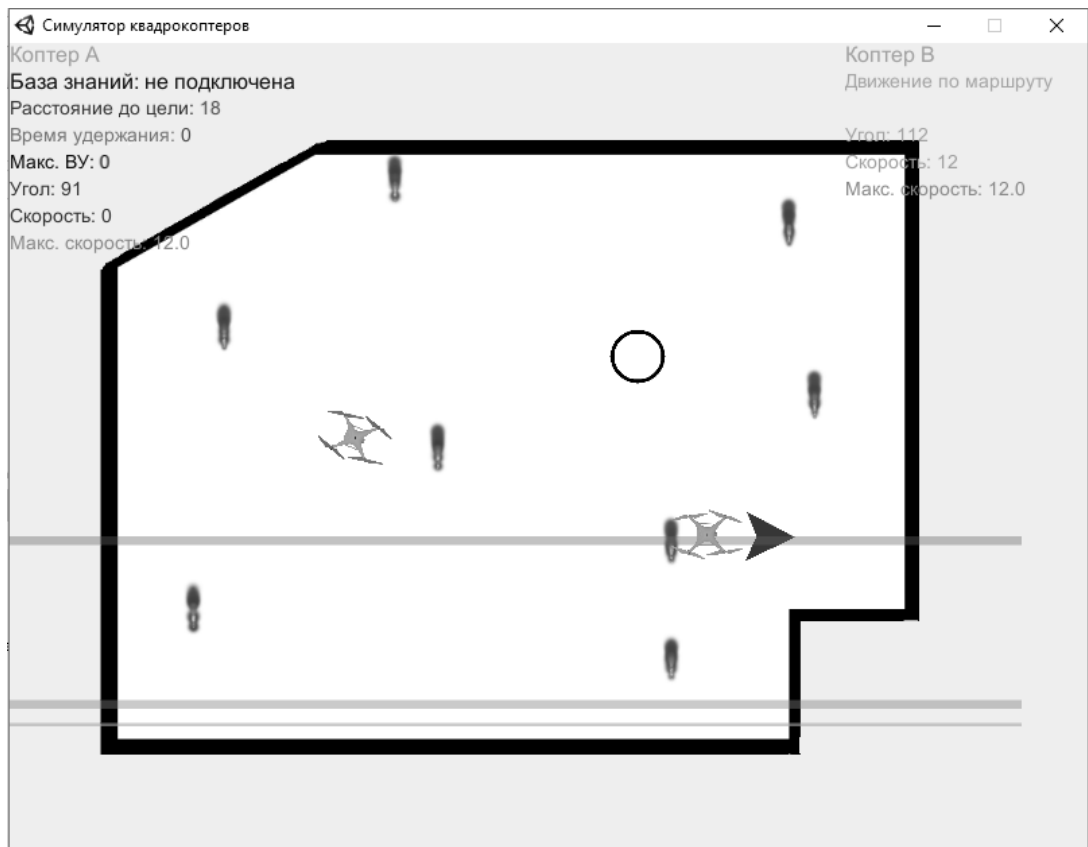


Рис. 4.13. Реализация виртуальной среды UNITY3D.

Исходное состояние системы

Сначала убегающий БПЛА летает в помещении, которое изображено на рисунке 4.13. База знаний пустая и подсистема ФРО реальных ситуаций в УС догоняющего БПЛА состоит только из ячеек, соответствующих 9-ти датчикам, которые являются базовыми образами (рис 4.14). А подсистема ФРО распознавания звуков была построена для распознавания двух слов «слева и справа» (рис 4.16). В аппарате эмоций записаны априорные оценки базовых образов (таб. 4.1).

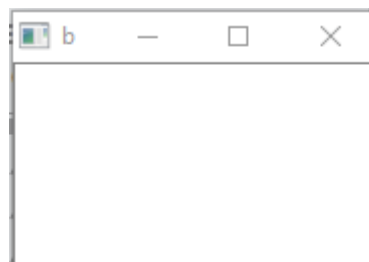


Рис. 4.14. Пустая база знаний

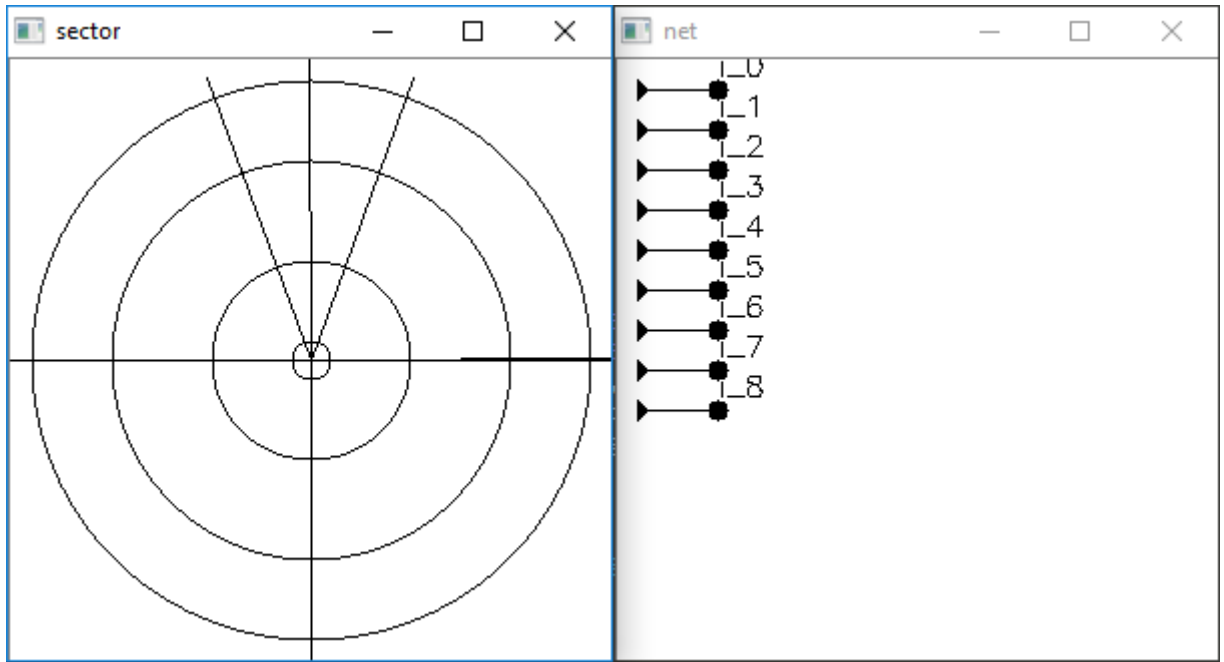


Рис. 4.15. Исходные состояния нейронной сети и сектора обзора БПЛА.

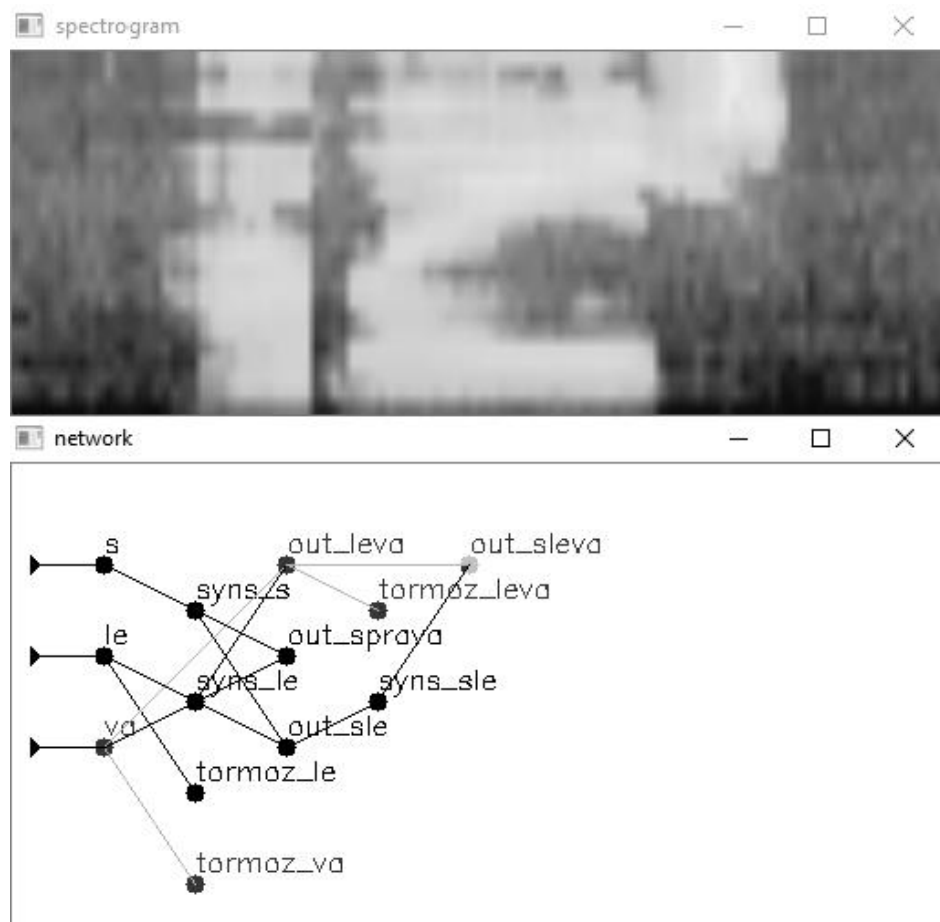


Рис. 4.16. Спектрограмма слова «слева» и фрагмент нейронной сети распознавания звуков. Слово «слева» было распознано.

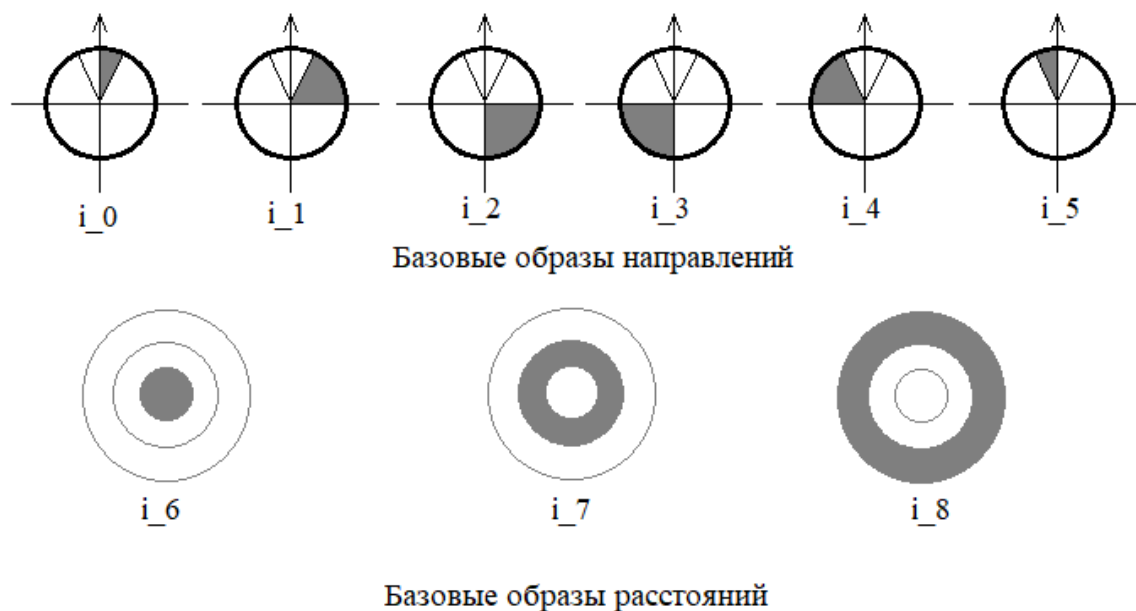


Рис. 4.17. Базовые образы в исходном состоянии системы управления.

Образ	i_0	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8
Оценка	10	5	1	1	5	10	10	5	1

Таб. 4.1 Таблица априорных оценок базовых образов.

Результат обучения системы

После запуска системы управления, база знаний пустая, догоняющий адаптивный БПЛА двигается по случайному направлению. При этом, его подсистема ФРО текущих ситуаций начинает автоматически находить и формировать закономерные образы (таб. 4.2). Подсистема ФРО реальных ситуаций представлена сетью из нейронов (рис. 4.18), при этом новые нейроны соответствуют новым ситуациям, которые адаптивный БПЛА выявил. Одновременно идут процессы обучения базы знаний (рис. 4.19), изменения

состояний нейронов и формирования новых образов. По мере того, как растет нейронная сеть, и как в базе знаний появляются новые знания, адаптивный БПЛА начинает летать преимущественно в сторону убегающего БПЛА и делать это все более точно.

Визуализация датчиков	Образ	Эмоциональная оценка
	$n_0 = i_0^{i_6}$	20
	$n_2 = i_5^{i_7}$	15
...
	$n_8 = i_3^{i_8}$	2
...

Таб. 4.2. Примеры сформированных образов и их оценки.

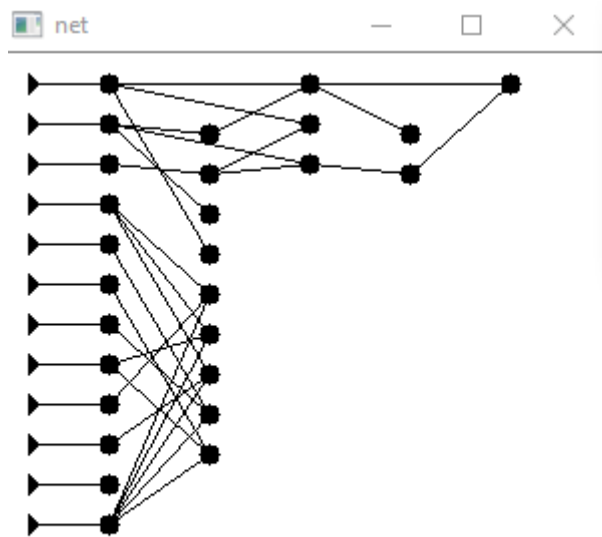


Рис. 4.18. Пример нейронной сети подсистемы ФРО, которая автоматически построилась через некоторое время процесса управления БПЛА.

На рисунке 4.19 изображена база знаний УС догоняющего БПЛА в виде таблицы. В ячейках записаны оценки для каждого события «условие & действие». значения оценок показаны цветом так, что чем выше оценка, тем более темный цвет закрашенной клетки.

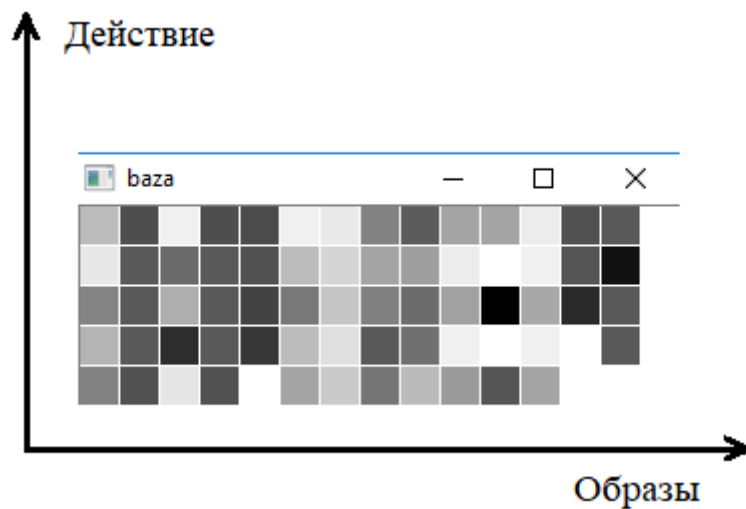


Рис. 4.19. Пример обучаемой базы знаний.

В рис. 4.20 показаны зависимости качества управления адаптивного БПЛА от наполненности базы знаний. Видно, что при пустой базе знаний, адаптивный БПЛА не может держать убегающий БПЛА в центральном секторе и не умеет его догонять. А когда база знаний заполнена, время задержки убегающего БПЛА в центральном секторе обзора увеличивается. Ниже, на рисунке 4.21 показано качество управления БПЛА, определенное по формуле 3.5.

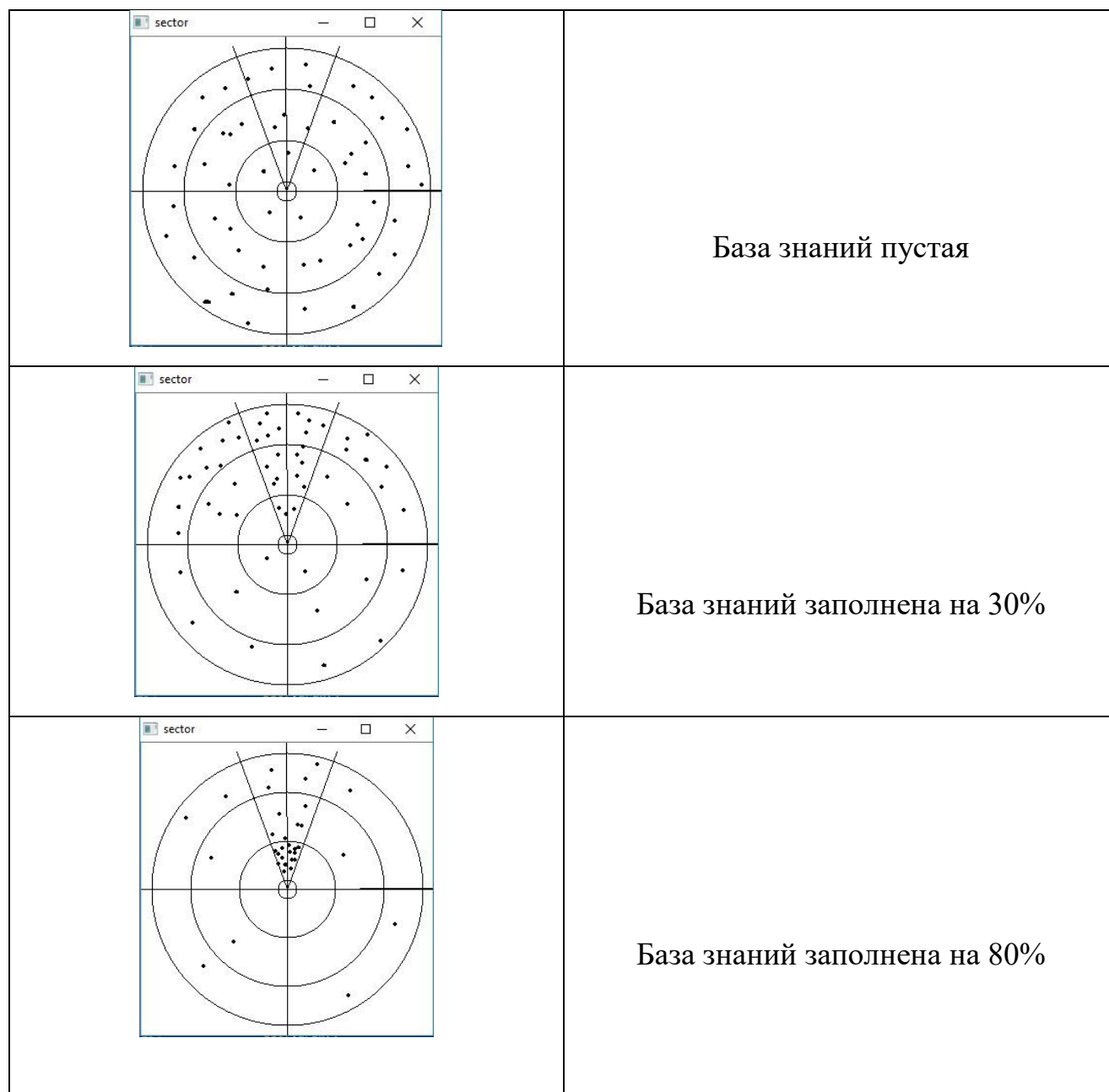


Рис 4.20. Зависимости качества управления от заполненности базы знаний.

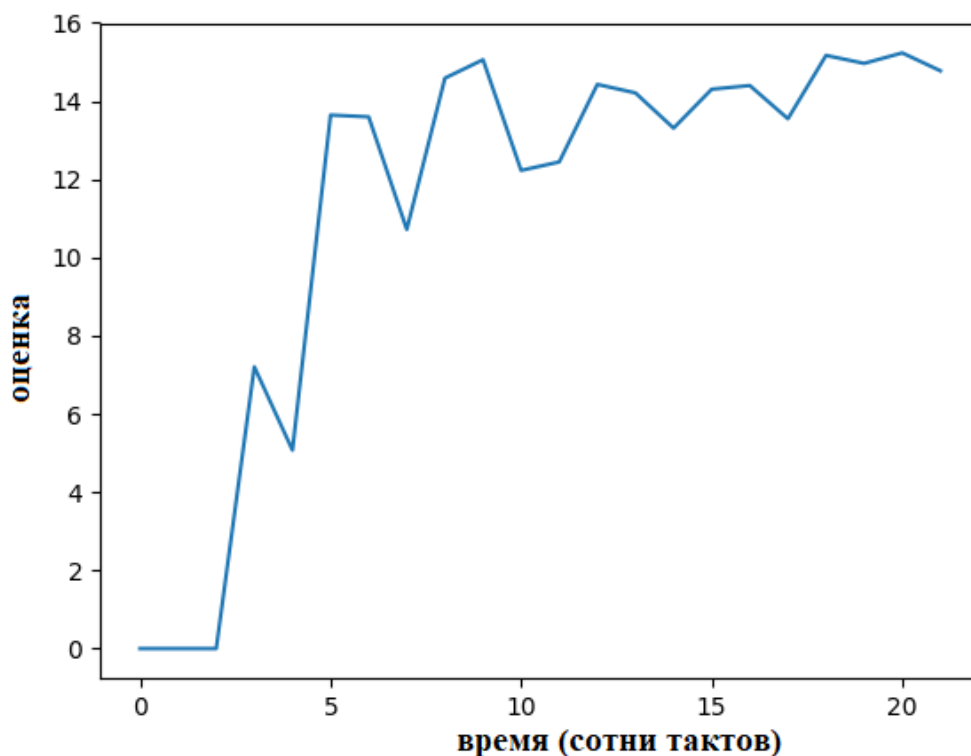


Рис. 4.21. Качество управления растет со временем по мере того, как догоняющий БПЛА адаптируется.

Чтобы показать адаптивные свойства метода ААУ, при достижении высокого качества управления, непосредственно во время процесса управления была симитирована «поломка оборудования БПЛА» - были поменяны местами значения двух датчиков, что изображено на рис. 4.22. При этом качество управление поначалу уменьшалось. Через некоторое время, база знаний адаптивного БПЛА автоматически переобучалась, и в результате высокое качество управления постепенно вновь восстанавливалось.

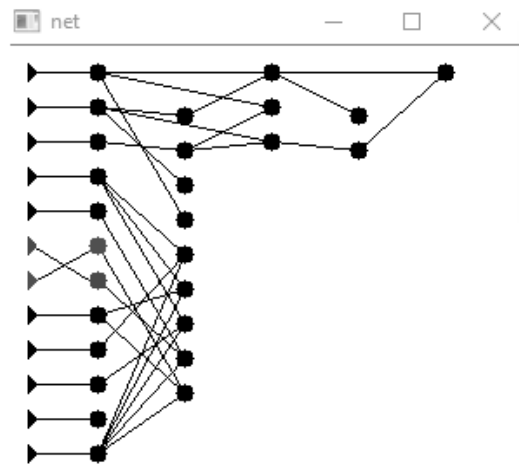


Рис. 4.22. Датчики поменялись местами в результате «поломки оборудования БПЛА».

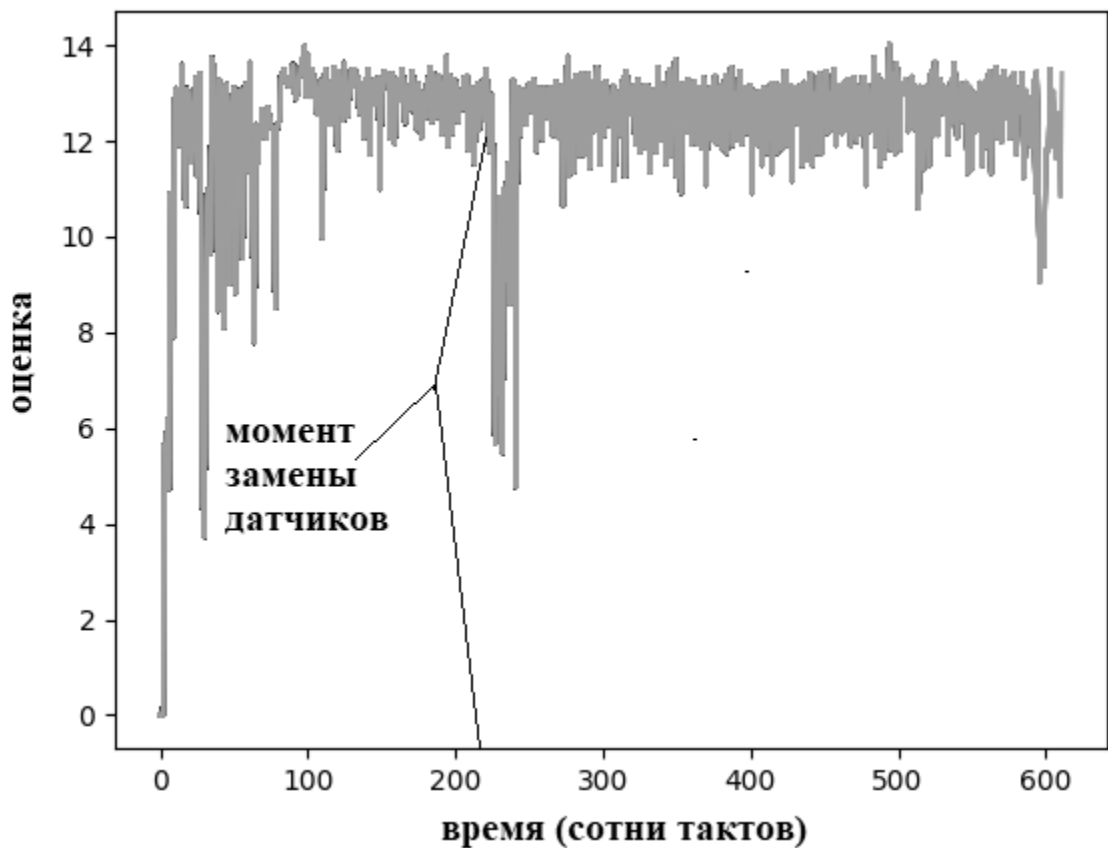


Рис. 4.23. Качество управления при замене датчиков, произошедшей вследствие имитации «поломки оборудования БПЛА». Качество управления в момент изменения системы датчиков упало, но через некоторое время система управления адаптируется к новым свойствам объекта управления.

Эксперимент с языковыми явлениями через голосовой интерфейс

Покажем результаты эксперимента, в котором демонстрируются языковые явления – способность управляющей системы ААУ автоматически ассоциировать реальные ситуации с одновременно произносимыми языковыми идентификаторами. В этом эксперименте, человек является учителем, который должен распознавать и озвучивать две ситуации, а именно, ситуацию, когда убегающий БПЛА (цель) оказывается слева от догоняющего БПЛА, и ситуацию, когда убегающий БПЛА (цель) оказывается справа от догоняющего БПЛА (рис. 4.18).

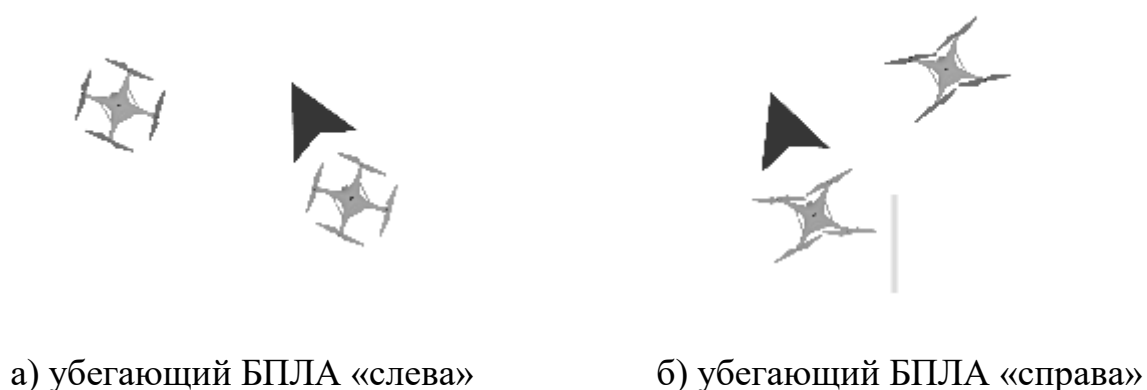


Рис. 4.24. Две ситуации для эксперимента.

При обнаружении одной из этих ситуаций, человек говорил слово «слева» или «справа», соответственно. При этом подсистемой ФРО догоняющего БПЛА образ реальной ситуации и звуковой образ распознавались одновременно. Поскольку это совпадение повторялось несколько раз, то в подсистеме ФРО из этих образов сформировался новый образ. Новорождённый образ является ассоциативным образом, у которого значение порога задано $\rho < 50\%$, чтобы образ мог быть распознан при наличии одного из этих прообразов.

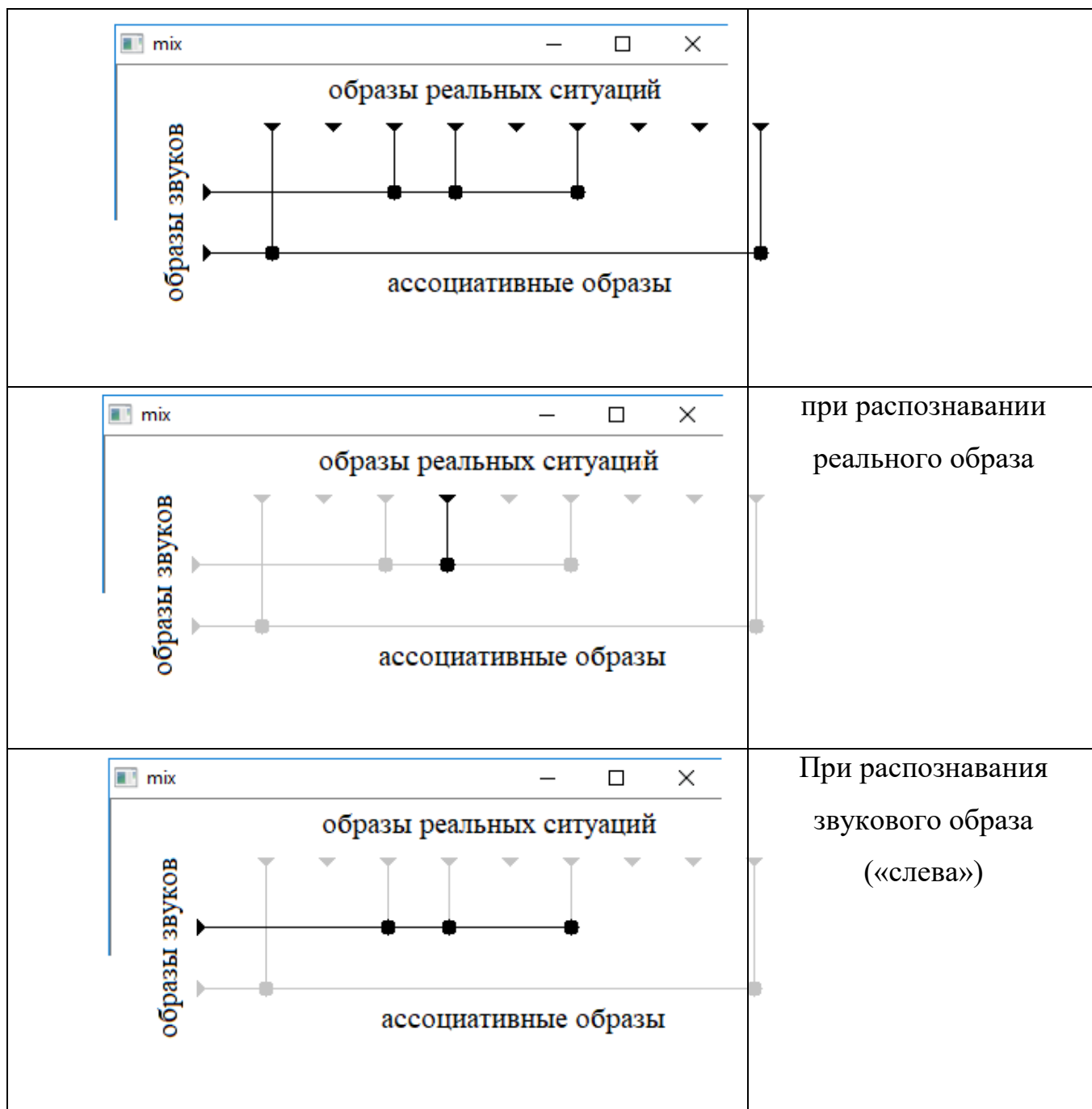


Рис. 4.25. Ассоциативный слой.

Тогда, в результате обучения, при отсутствии звуковых сигналов адаптивный БПЛА хорошо догонял убегающий БПЛА, как обычно. При отключении потока входной информации в подсистему ФРО от реальных образов, адаптивный БПЛА только стоял на месте. При этом мы говорим слово «слева», если убегающий БПЛА находится в левых секторах обзора адаптивного БПЛА, и

мы говорим слово «справа», если убегающий БПЛА находится в правых секторах обзора адаптивного БПЛА. После получения звукового идентификатора, БПЛА начинал реагировать и понимать реальную ситуацию, которой он обучился. Качество управления только с помощью языковых подсказок, естественно, ниже, чем при использовании визуальных датчиков, но это так же, как в жизни – можно управлять летящим «вслепую» пилотом с помощью звуковых подсказок, качество управления не может быть очень высоким, поскольку нет такого количества слов, чтобы идентифицировать все возможные ситуации, но это может сыграть свою роль, если у пилота нет возможности самому видеть цель, или он ее по каким-то причинам не замечает, но он может услышать подсказки со стороны.

Некоторые полезные приложения нейроноподобных сетей из метода ААУ

В дополнение рассмотрим несколько возможных приложений нейроноподобной реализации системы ААУ для решения некоторых задач анализа данных.

Приложение 1 – система распознавания рукописных цифр MNIST [41]

MNIST это база рукописных цифр, которая часто используется для тестирования алгоритмов распознавания. База MNIST состоит из 70000 цифр, 60000 из которых используются для обучения, а остальные цифры для тестирования.

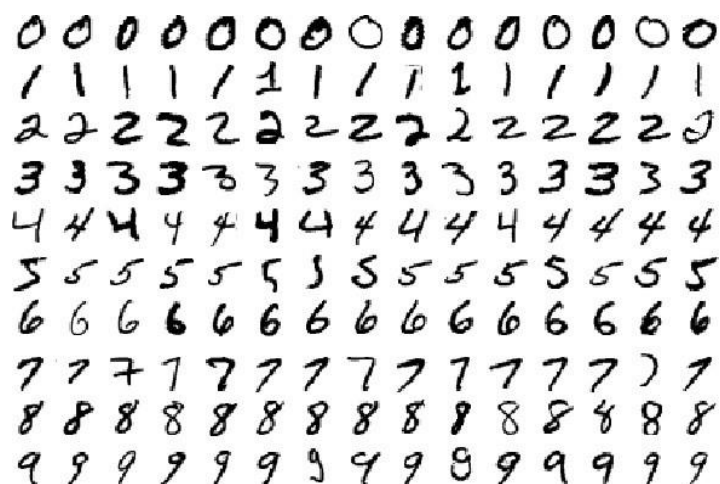


Рис. 5.1. Часть базы рукописных цифр.

Данные картины цифр представляют собой картину размерностей 28x28 пикселей. Поэтому рецепторное поле также состоит из матрицы рецепторов размера 28x28.

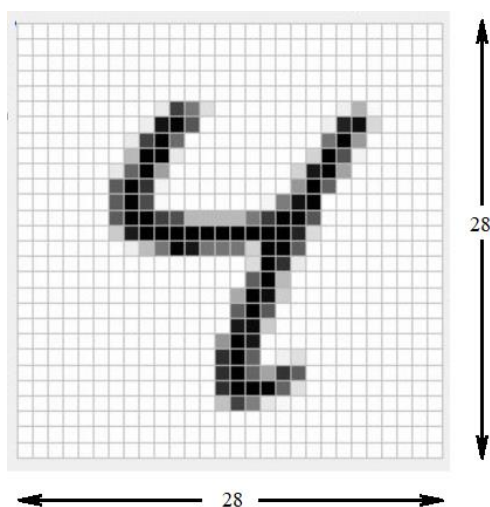


Рис. 5.2. Пример обучаемой цифры 4.

Чтобы получить первую стандартную библиотеку фильтров, мы разбиваем все картины цифр окном фиксированного размера (пример 7x7) с перекрытием (рис. 5.3).

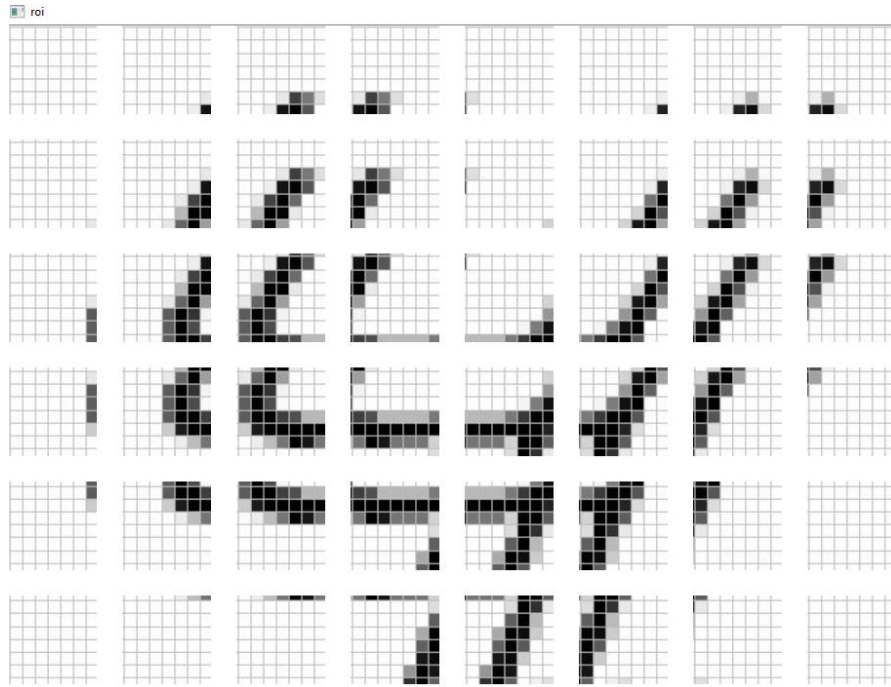


Рис. 5.3. Разделение цифры окнами фиксированного размера.

В результате этого этапа мы получили набор частей цифр, из которых выбираем 12-е самых популярных частей для построения стандартной библиотеки фильтров.

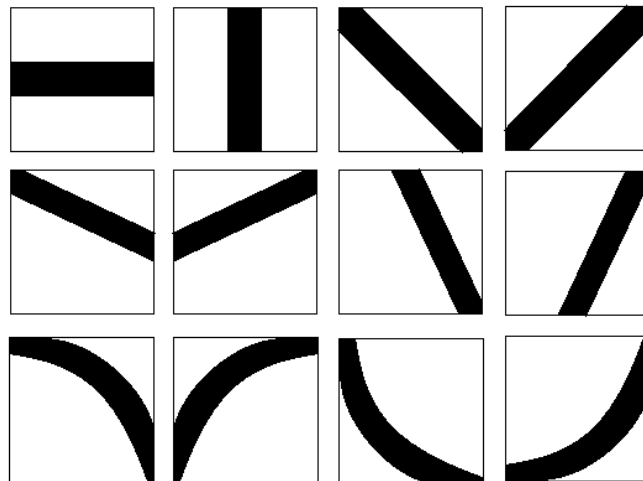


Рис. 5.4. Стандартная библиотека фильтров размерности 7x7.

Из картин библиотеки фильтров строятся нейронные фильтры на основе нейронов 1-го типа (рис. 5.5).

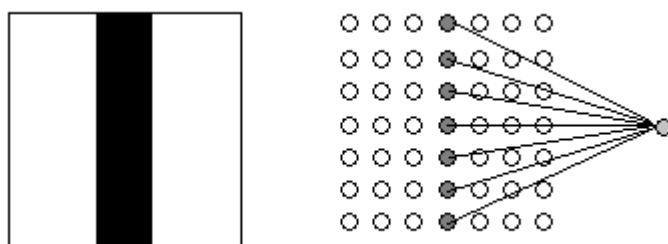


Рис. 5.5. Нейронная конструкция одного фильтра.

Нейронные фильтры перемещаются по рецептивному полю. В результате получаем карты признаков, соответствующие фильтрам. Например, см. рис. 5.6.

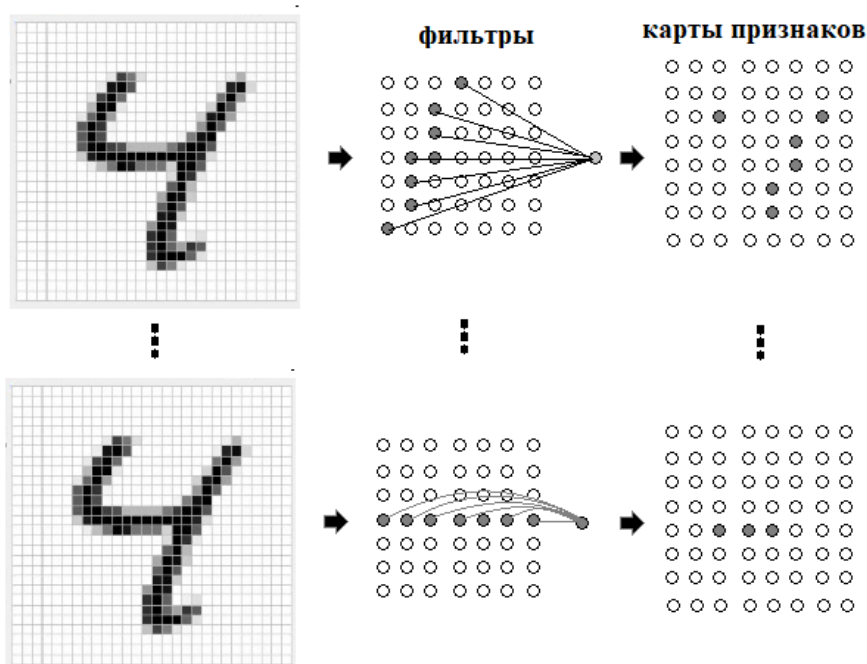


Рис. 5.6. Результаты фильтрования.

Нейрон класса цифры создается соединением всех активных нейронов карт признаков (рис. 5.7).

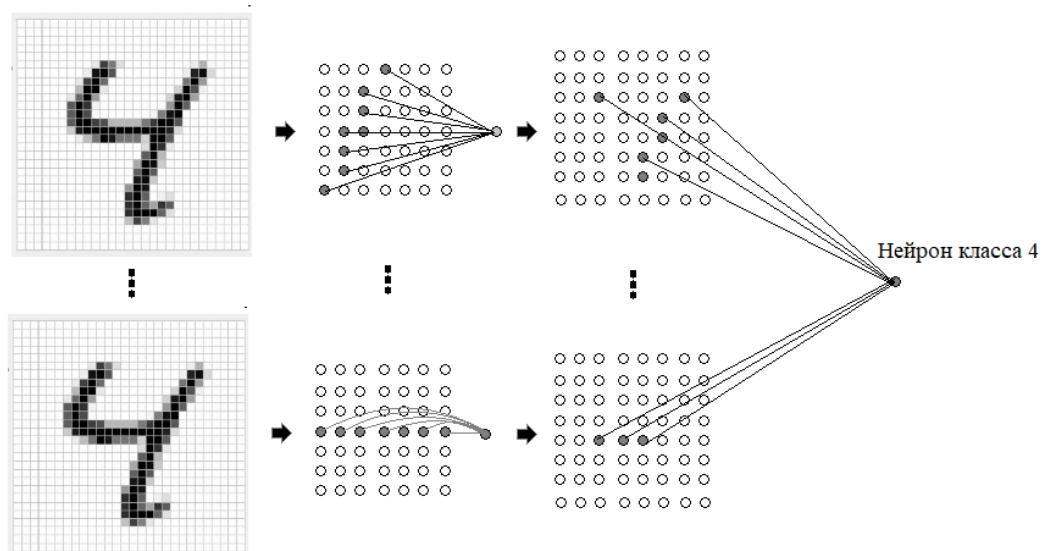


Рис. 5.7. Процесс формирования класса цифр 4.

Это все этапы процесса генерации подсети для одного образа цифры. Процесс обучения – это процесс генерации структуры нейронной сети. При поступлении всех обучаемых цифр, получаем полную нейронную сеть, которая используется для распознавания 10 цифр (от 0 до 9). Нейронная сеть изображена рис. 5.8.

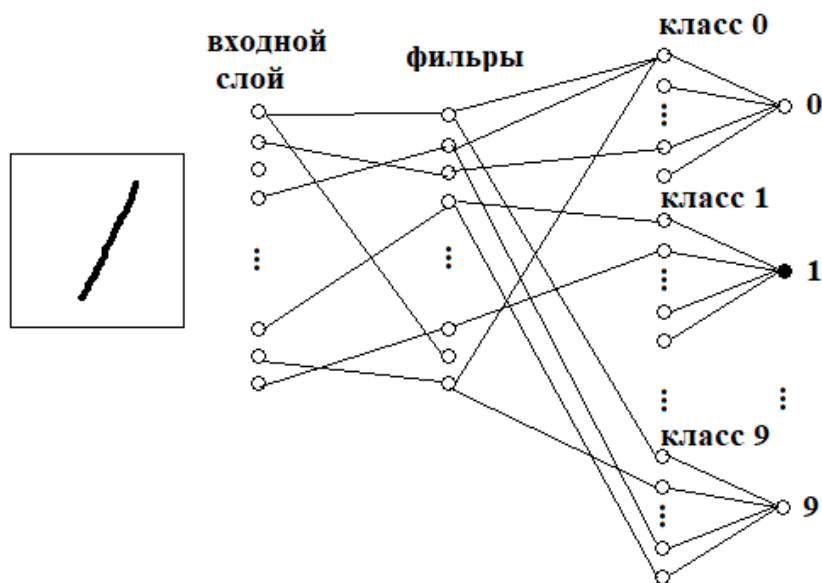


Рис. 5.8. Нейронная сеть неполной связи для распознавания цифр.

При обучении 60.000 рукописных цифр, в сети были сгенерированы около 6.000 нейронов. Качество распознавания цифр достигло 97,5%.

Приложение 2 – Кластеризация на основе нейроноподобных сетей системы ААУ

Алгоритм кластеризации — это задача разбиения множества элементов на группы по определенным признакам.

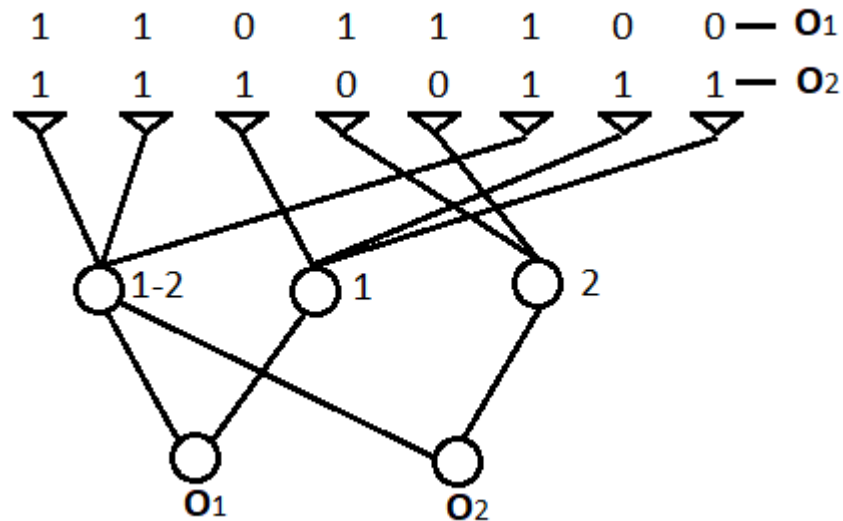
Существует несколько алгоритмов, позволяющих решать задачу кластеризации, таких как: К-Средних (K-means)[42], нейронной сети self-organization map (SOM) [43]. В этом предложении мы представляем новый подход к задаче кластеризации, который основан на нейроноподобных сетях ААУ [5]. Процесс обучения нейроноподобной сети [5] для кластеризации является процессом построения структуры сети.

Определение 1: Уровень сходства (c) - это количество общих признаков;

Определение 2: Количество объектов подкласса (δ) определяет отношение похожих объектов к определенному сходству;

Процесс обучения состоит из следующих шагов:

- Формирование образов при поступлении информации прообразов с разделением сходства и отличия.



- Случайно выбираем объект и смотрим сходства у него. Если у данного сходства количество других объектов, находящихся в окрестности, больше чем δ , то мы создаем класс. Если количество других объектов, находящихся в окрестности, меньше, чем δ , то мы повторяем шаг 2.
- Если класс сформирован, то мы переходим к другому объекту, находящемуся в данном подклассе. Далее определяем уровень сходства и количество похожих объектов в его собственном подклассе. Если это количество больше, чем δ , то добавляем в сформированный класс. Если нет, то переходим в шаг 2.

Этот метод был применен для автоматической кластеризации рукописных цифр MNIST [41]. в результате мы получим классы, у которых уровень сходства (с) находится в пределах 70%-75%.

СПИСОК ЛИТЕРАТУРЫ

1. Жданов А.А., "О понятии автономного искусственного интеллекта." Сб. научн. тр. Искусственный интеллект в технических системах.-М.: Гос. ИФТП (1997).
2. Жданов А.А. "Метод автономного адаптивного управления." Известия Южного федерального университета. Технические науки 38.3 (2004).
3. Жданов А.А., "Формальная модель нейрона и нейросети в методологии автономного адаптивного управления." Сб. " Вопросы кибернетики". Научный совет по комплексной проблеме " Кибернетика" РАН 3 (1997).
4. Жданов А.А., Т. С. Наумкина. "Моделирование языковых явлений в нейроноподобных системах управления." Сб. научн. тр. Всероссийской научно-технической конференции" Нейроинформатика-2007", ч. Vol. 3.
5. Жданов А.А., Автономный искусственный интеллект, М.: «Бином. Лаборатория знаний», 2008.
6. Weizenbaum, Joseph. "ELIZA—a computer program for the study of natural language communication between man and machine." *Communications of the ACM* 9.1 (1966): 36-45.
7. Нильсон, Нильс. Искусственный интеллект. Рипол Классик, 1973.
8. Hubel, David H. *Eye, brain, and vision*. Scientific American Library/Scientific American Books, 1995.
9. Налимов, Василий Васильевич. Вероятностная модель языка. О соотношении естественных и искусственных языков. Рипол Классик, 1974.
10. Кузнецов, С. Н. "Международные языки; Искусственные языки." *Лингвистический энциклопедический словарь*(1990).
11. Виноград, Т. "Программа, понимающая естественный язык." (1976).
12. Пиотровский, Раймонд Генрихович. Текст, машина, человек. Наука, Ленингр. отд-ние, 1975.

13. Bahdanau, Dzmitry, Kyunghyun Cho, and YoshuaBengio. "Neural machine translation by jointly learning to align and translate." arXiv preprint arXiv:1409.0473 (2014).
14. Zhang, Yu, William Chan, and Navdeep Jaitly. "Very deep convolutional networks for end-to-end speech recognition." Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on. IEEE, 2017.
15. Л.В. Нейман, М. Р. Богомильский, В. И. Селиверстов. Анатомия, физиология и патология органов слуха и речи. М.: Владос, 2001.
16. Muda, Lindasalwa, MumtajBegam, and IrraivanElamvazuthi. "Voice recognition algorithms using mel frequency cepstral coefficient (MFCC) and dynamic time warping (DTW) techniques." arXiv preprint arXiv:1003.4083 (2010).
17. Qiu, Minghui, et al. "Alime chat: A sequence to sequence and rerank based chatbot engine." Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). Vol. 2. 2017.
18. Weizenbaum, Joseph. "ELIZA—a computer program for the study of natural language communication between man and machine." Communicationsofthe ACM 9.1 (1966): 36-45.
19. Chatbot Ipal
20. Heal, Jane. "Expression and Meaning: Studies in the Theory of Speech Acts By Searle John R. Cambridge University Press, 1979, xiv+ 187 pp.,£ 8.50." Philosophy 56.216 (1981): 270-271.
21. Searle, John R. "Minds, brains, and programs." Behavioralandbrainsciences 3.3 (1980): 417-424.
22. Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In Advances in neural information processing systems, pages 3104–3112, 2014.
23. Цыпкин Я.З. Адаптация и обучение в автоматических системах // М.: Наука, 1968.
24. Афанасьев В.Н., Колмоновский В.Б., Носов В.Р. Математическая теория конструирования систем управления.// М.: «Высшая школа», 1998.

25. Круглов, Владимир Васильевич, М. И. Дли, and Р. Ю. Голунов. Нечеткая логика и искусственные нейронные сети. М.: Физматлит, 2001.
26. Рыбина, Галина Валентиновна. "Обучающие интегрированные экспертные системы: некоторые итоги и перспективы." Искусственный интеллект и принятие решений 1 (2008): 22-46.
27. Кибрик, А. Е. "Язык//Лингвистический энциклопедический словарь." Москва (1990): 604-606.
28. Рябцева, Надежда Константиновна. Язык и естественный интеллект. Academia, 2005.
29. Bartl, Alexander, and Gerasimos Spanakis. "A retrieval-based dialogue system utilizing utterance and context embeddings." arXiv preprint arXiv:1710.05780 (2017).
30. <https://www.uantwerpen.be/en/research-groups/op3mech/projects---publicati/academic/project-aslan/>
31. Хьюбел, Дэвид. Глаз, мозг, зрение. Vol. 239. М.: мир, 1990.
32. Graves A, Mohamed AR, Hinton G. Speech recognition with deep recurrent neural networks. In Acoustics, speech and signal processing (icassp), 2013 IEEE international conference on 2013 May 26 (pp. 6645-6649). IEEE.
33. Hoffmann, G., 2011. CIE Lab color space (2003).
34. Нуссбаумер, Генри. "Быстрое преобразование Фурье и алгоритмы вычисления сверток." (1985).
35. Л.В. Нейман, М. Р. Богомильский, В. И. Селиверстов. Анатомия, физиология и патология органов слуха и речи. М.: Владос, 2001
36. Muda, Lindasalwa, Mumtaj Begam, and Irraivan Elamvazuthi. "Voice recognition algorithms using mel frequency cepstral coefficient (MFCC) and dynamic time warping (DTW) techniques." arXiv preprint arXiv:1003.4083 (2010).
37. А.А. Жданов, Т.С. Наумкина, Моделирование языковых явлений в нейроноподобных системах управления // Сб. научн. тр. Всероссийской научно-технической конференции "Нейроинформатика-2007", ч. 3, стр. 76-84.

38. Alexander Zhdanov, Alexander Kondukov, Tamara Naumkina, Olga Dmitrienko, Automatic origin of a language in AAC neuron-like systems: Proceedings of the 11th International Conference "Speech and Computer SPECOM'2006", pages 550 - 554.
39. A.A. Zhdanov, T.S. Naumkina, Modeling of Formation of Extralinguistic Factors Influencing the Attitude of the Recipient towards Language Messages. ("Моделирование формирования экстралингвистических факторов, влияющих на отношение реципиента к языковым сообщениям"): The 12th International Conference SPECOM-2007 Proceedings. P. 833-838.
40. Aur, Dorian. "Where is the 'Jennifer Aniston neuron'?" (2010).
41. LeCun, Yann, Corinna Cortes, and Christopher JC Burges. "MNIST handwritten digit database." AT&T Labs [Online]. Available: <http://yann. lecun. com/exdb/mnist> 2 (2010).
42. Hartigan, John A., and Manchek A. Wong. "Algorithm AS 136: A k-means clustering algorithm." Journal of the Royal Statistical Society. Series C (Applied Statistics) 28.1 (1979): 100-108.
43. Kohonen, Teuvo. "The self-organizing map." Proceedings of the IEEE 78.9 (1990): 1464-1480.
44. Нгуен Нгок Зиеп, Мань Ха Ле. "Нейросетевой метод снятия омонимии." Труды Московского физико-технического института 7.4 (2015): 174-182.
45. Чан Ван Нгиа, Ву Вьет Тханг, Нгуен Нгок Зиеп, реализация нейронной сети на fpga для распознавания атаки на информационные системы, II Международная конференция «Инжиниринг & телекоммуникации En&T 2015».
46. Степанян И.В., Зиеп Н.Н. Растущие свёрточные нейроподобные структуры для задач распознавания статических образов // Нейрокомпьютеры: разработка, применение. 2018. № 5. С. 5-14.
47. Зиеп, Н.Н., Жданов, А.А. НЕЙРОПОДОБНЫЙ ПОДХОД К РАСПОЗНАВАНИЮ РЕЧИ. Программирование, 2018 (3), pp.49-62.

48. Нгуен Нгок Зиеп, Мишустин И.А. Принцип системы распознавания образов на основе нейроноподобных сетей. IV Международная конференция «Инжиниринг & телекоммуникации En&T 2017» С. 53-54.
49. Жданов А.А., Нгуен Н.З., Перский Г.С., Пешенко Р.Э., Пижонков А.Г., Степанян И.В., Сямиуллин З.С., "программа динамического синтеза растущих биоподобных структур на основе искусственных нейронов с расширенной функциональностью и способностью к дообучению для задач распознавания образов и управления динамическими объектами". Свидетельство о регистрации программ на ЭВМ РФ № 2018661458.
50. Крыжановский, Михаил Владимирович. Принципы нейроноподобной реализации систем Автономного Адаптивного Управления. Диссертация. Ин-т систем. программирования Рос. акад. наук, 2004.
51. Крыжановский М.В., Магомедов Б.М. Модифицированный стековый алгоритм как модель подсистемы формирования и распознавания образов мобильного робота. // Ж. Искусственный интеллект. 2004. 3. С. 320-328.