

Школа: ИШИТР Инженерная школа информационных технологий и робототехники
 Направление подготовки 09.03.01 «Информатика и вычислительная техника»
 Отделение школы (НОЦ): Отделение информационных технологий

БАКАЛАВРСКАЯ РАБОТА

Тема работы
Разработка прогнозных моделей на базе искусственных нейронных сетей

УДК 004.415:551.509:004.85

Студент

Группа	ФИО	Подпись	Дата
3-8В41	Шкель Никита Иванович		

Руководитель ВКР

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ	Ботыгин И.А.	к.т.н., Доцент		

КОНСУЛЬТАНТЫ ПО РАЗДЕЛАМ:

По разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОСГН	Криницына З.В.	к.т.н., Доцент		

По разделу «Социальная ответственность»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Ассистент ООД	Мезенцева И.Л.	ассистент		

ДОПУСТИТЬ К ЗАЩИТЕ:

Руководитель ООП	ФИО	Ученая степень, звание	Подпись	Дата
09.03.01 Информатика и ВТ	Погребной А. В.	к.т.н., Доцент		

**ЗАПЛАНИРОВАННЫЕ РЕЗУЛЬТАТЫ ПО ОСНОВНОЙ
ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЕ ПОДГОТОВКИ БАКАЛАВРОВ
09.03.01 «ИНФОРМАТИКА И ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА», НИ ТПУ,
ПРОФИЛЬ «ВЫЧИСЛИТЕЛЬНЫЕ МАШИНЫ, КОМПЛЕКСЫ, СИСТЕМЫ И
СЕТИ»**

Код результатов	Результат обучения (выпускник должен быть готов)
<i>Профессиональные компетенции</i>	
P1	Применять базовые и специальные естественнонаучные и математические знания в области информатики и вычислительной техники, достаточные для комплексной инженерной деятельности.
P2	Применять базовые и специальные знания в области современных информационных технологий и решения инженерных задач.
P3	Ставить и решать задачи комплексного анализа, связанные с созданием аппаратно-программных средств информационных и автоматизированных систем, с использованием базовых и специальных знаний, современных аналитических методов и моделей.
P4	Разрабатывать программные и аппаратные средства (системы, устройства, блоки, программы, базы данных и т.п.) в соответствии с техническим заданием и с использованием средств автоматизации проектирования.
P5	Проводить теоретические и экспериментальные исследования, включающие поиск и изучение необходимой научно-технической информации, математическое моделирование, проведение эксперимента, анализ и интерпретация полученных данных, в области создания аппаратных и программных средств информационных и автоматизированных систем.
P6	Внедрять, эксплуатировать и обслуживать современные программно-аппаратные комплексы, обеспечивать их высокую эффективность, соблюдать правила охраны здоровья, безопасность труда, выполнять требования по защите окружающей среды.
<i>Универсальные компетенции</i>	
P7	Использовать базовые и специальные знания в области проектного менеджмента для ведения комплексной инженерной деятельности.
P8	Владеть иностранным языком на уровне, позволяющем работать в иноязычной среде, разрабатывать документацию, презентовать и защищать результаты комплексной инженерной деятельности.
P9	Эффективно работать индивидуально и в качестве члена группы, состоящей из специалистов различных направлений и квалификаций, демонстрировать ответственность за результаты работы и готовность следовать корпоративной культуре организации.
P10	Демонстрировать знания правовых, социальных, экономических и культурных аспектов комплексной инженерной деятельности.
P11	Демонстрировать способность к самостоятельному обучению в течение всей жизни и непрерывному самосовершенствованию в инженерной профессии.

Министерство науки и высшего образования Российской Федерации
 федеральное государственное автономное
 образовательное учреждение высшего образования
 «Национальный исследовательский Томский политехнический университет» (ТПУ)

Школа ИШИТР Инженерная школа информационных технологий и робототехники
 Направление подготовки (специальность) 09.03.01 «Информатика и вычислительная техника»
 Отделение школы (НОЦ) Отделение информационных технологий

УТВЕРЖДАЮ:
 Руководитель ООП

 (Подпись) (Дата) (Ф.И.О.)

ЗАДАНИЕ **на выполнение выпускной квалификационной работы**

В форме:

Бакалаврской работы

Студенту:

Группа	ФИО
38В41	Шкель Никита Иванович

Тема работы:

Разработка прогнозных моделей на базе искусственных нейронных сетей	
Утверждена приказом директора (дата, номер)	От 11.03.2019 №1799/с

Срок сдачи студентом выполненной работы:	
--	--

ТЕХНИЧЕСКОЕ ЗАДАНИЕ:

Исходные данные к работе	Техническое задание к реализации разработки программного обеспечения для прогнозирования основных метеорологических параметров с использованием методов машинного обучения.
Перечень подлежащих исследованию, проектированию и разработке вопросов	Изучение документации о создании приложения; Выбор проектных решений и инструментов для реализации приложения; Реализация приложения с использованием выбранного инструментария; Тестирование и отладка созданного ПО; Расчет ресурсэффективности и ресурсосбережения и анализ вредных производственных факторов.

Перечень графического материала		Пример дизайна и работы ПО
Консультанты по разделам выпускной квалификационной работы		
Раздел	Консультант	
Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	Доцент отделения социально-гуманитарных наук Криницына З. В.	
Социальная ответственность	Ассистент отделения общетехнических дисциплин Мезенцева И.Л.	

Дата выдачи задания на выполнение выпускной квалификационной работы по линейному графику	
---	--

Задание выдал руководитель):

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент отделения информационных технологий	Ботыгин Игорь Александрович	К.т.н. Доцент		

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
38B41	Шкель Никита Иванович		

**ЗАДАНИЕ ДЛЯ РАЗДЕЛА
«ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И
РЕСУРСОСБЕРЕЖЕНИЕ»**

Студенту:

Группа	ФИО
3-8В41	Шкель Никите Ивановичу

Школа	ИШИТР	Отделение школы (НОЦ)	ОИТ
Уровень образования	Бакалавриат	Направление/специальность	09.03.01

Исходные данные к разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»:

1. Стоимость ресурсов научного исследования (НИ): материально-технических, энергетических, финансовых, информационных и человеческих	Стоимость НИИ – 215274,9 руб. Зарботная плата – 97330 руб. Материальные затраты – 6850 руб.
2. Нормы и нормативы расходования ресурсов	Предельная годовая норма амортизации оборудования 20%
3. Используемая система налогообложения, ставки налогов, отчислений, дисконтирования и кредитования	Отчисления по страховым взносам – 27,1% от ФОТ

Перечень вопросов, подлежащих исследованию, проектированию и разработке:

1. Оценка коммерческого потенциала, перспективности и альтернатив проведения НИ с позиции ресурсоэффективности и ресурсосбережения	SWOT-анализ.
2. Планирование и формирование бюджета научных исследований	SWOT-анализ.
3. Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования	Формирование бюджета проекта.

Перечень графического материала (с точным указанием обязательных чертежей):

1. Оценка конкурентоспособности технических решений
2. Матрица SWOT
3. Альтернативы проведения НИ
4. График проведения и бюджет НИ
5. Оценка ресурсной, финансовой и экономической эффективности НИ

Дата выдачи задания для раздела по линейному графику	
--	--

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОСГН ШБИП	Криницына Зоя Васильевна	К.т.н Доцент		

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
3-8В41	Шкель Никита Иванович		

ЗАДАНИЕ ДЛЯ РАЗДЕЛА «СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ»

Студенту:

Группа	ФИО
3-8B41	Шкель Никите Ивановичу

Школа	Информационных технологий и робототехники	Отделение школы (НОЦ)	Информационных технологий
Уровень образования	Бакалавриат	Направление/специальность	Информатика и вычислительная техника

Исходные данные к разделу «Социальная ответственность»:

1. Характеристика объекта исследования (вещество, материал, прибор, алгоритм, методика, рабочая зона) и области его применения	Объектом исследования является программное обеспечение. Данное программное обеспечение служит для прогнозирования основных метеорологических параметров с использованием методов машинного обучения Рабочее место – РМ с персональными компьютерами.
--	---

Перечень вопросов, подлежащих исследованию, проектированию и разработке:

1. Правовые и организационные вопросы обеспечения безопасности: – специальные (характерные при эксплуатации объекта исследования, проектируемой рабочей зоны) правовые нормы трудового законодательства; – организационные мероприятия при компоновке рабочей зоны..	"Гигиенические требования к микроклимату производственных помещений" СанПиН 2.2.4.548-96 "Шум на рабочих местах, в помещениях жилых, общественных зданий и на территории жилой застройки" СН 2.2.4/2.1.8.562-96 "Гигиенические требования к естественному, искусственному и совмещенному освещению жилых и общественных зданий" СанПиН 2.2.1/2.1.1.1278-03 Освещение рабочих мест внутри зданий. Нормы и методы измерений ГОСТ Р 55710-2013 СП 52.13330.2016 Естественное и искусственное освещение. Актуализированная редакция СНиП 23-05-95* Требования по пожарной безопасности ГОСТ 12.1.004–91
2. Производственная безопасность: 2.1. Анализ выявленных вредных и опасных факторов 2.2. Обоснование мероприятий по снижению воздействия	• Отклонение показателей микроклимата • Недостаточная освещенность рабочей зоны • Повышенный уровень шума на рабочем месте • Опасность поражения электрическим током
3. Экологическая безопасность:	Анализ негативного воздействия на окружающую природную среду: утилизация люминесцентных ламп, компьютеров и другой оргтехники
4. Безопасность в чрезвычайных ситуациях:	Возникновение пожара

Дата выдачи задания для раздела по линейному графику

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
ассистент ООТД	Мезенцева И.Л.	ассистент		

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
3-8B41	Шкель Никита Иванович		

РЕФЕРАТ

Пояснительная записка содержит 118 страниц, 36 рисунков, 19 таблиц и 107 источников.

Данная квалификационная работа посвящена разработке программного обеспечения для прогнозирования основных метеорологических параметров, с использованием методов машинного обучения. Многие функции реализованы на основе изучения предметной области и аналогов. Разработан дружественный интерфейс пользователя, удовлетворяющий требованиям проекта.

Ключевые слова: нейросети, метеорология, прогноз.

Объектом исследования является разработка программного обеспечения.

Цель работы - разработка прогнозных моделей на базе искусственных нейронных сетей.

В процессе исследования проводились анализы возможностей реализации данного проекта.

В результате исследования был выбран язык программирования C++, алгоритм, на основе фильтров Хебба, сети SOM.

Область применения: Прогнозирование основных метеорологических параметров.

ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ, СОКРАЩЕНИЯ И НОРМАТИВНЫЕ ССЫЛКИ

МД - метеорологические данные

МП - метеорологические параметры

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	10
1. ИССЛЕДОВАТЕЛЬСКИЙ РАЗДЕЛ	13
1.1. Прогнозирование метеорологических прогнозов. Современное состояние	13
1.2. Анализ средств и методов автоматизации краткосрочного прогнозирования	22
1.3. Модель подготовки данных для прогнозирования метеорологических параметров	26
2. СПЕЦИАЛЬНЫЙ РАЗДЕЛ	32
2.1. Обработка многомерного сигнала фильтром Хебба	32
2.2. Вычислительный алгоритм предварительной подготовки данных	36
2.3. Самоорганизующиеся системы и обучение сети	37
2.4. Описание алгоритма для выявления значимых метеорологических параметров	42
2.5. Компьютерное моделирование и анализ нейронной сети	43
2.6. Алгоритм квантования вектора обучения	48
2.7. Адаптивная классификация множеств	51
2.8. Иерархическое квантование векторов	54
2.9. Отображение результатов анализа	60
3. ТЕХНОЛОГИЧЕСКИЙ РАЗДЕЛ	65
3.1. Информационное обеспечение краткосрочного прогнозирования метеорологических параметров	65
3.2. Техническое обеспечение системы прогнозирования	65
3.3. Программная реализация подготовки коэффициентов	66
3.4. Программная реализация нейронной сети для кластеризации метеорологических данных	74
3.5. Тестирование комплекса	80
4. ПРОИЗВОДСТВЕННАЯ И ЭКОЛОГИЧЕСКАЯ БЕЗОПАСНОСТЬ	83
4.1 Описание Рабочего места оператора	83
4.2 Мероприятия по оптимизации рабочего места оператора	84
4.3 Расчет освещения рабочего места	90
5. ЭКОНОМИЧЕСКИЙ РАЗДЕЛ	94
5.1 Выбор и обоснование методики расчета экономической эффективности	94
5.2 Расчет показателей экономической эффективности проекта	96
5.3 Расчет величины потерь (рисков)	106
ЗАКЛЮЧЕНИЕ	108
СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ	109
Приложение 1. Исходный код	119
Приложение 2. Форматы хранения метеорологических данных	139

ВВЕДЕНИЕ

Для построения прогноза основных метеорологических параметров (МП), в научных работах отечественных и зарубежных метеорологов исследовалась корреляция метеорологических данных (МД). В качестве метеорологических параметров (МП) для оценки корреляции рассматривались послойная температура атмосферы, давление, скорость ветра, влажность, содержание озона, прозрачность воздушной среды, распределение и состав облачного слоя, температура почвы, радиационного фона [1-9] и т.д.

Актуальность задачи. Основной трудностью реализации прогноза в существующих методах и подходах является, во-первых, отсутствие строгой математической модели корреляции характеристик МД, между собой, и их корреляции с событием изменения какого-либо МП, во-вторых сложность выбора и анализа именно тех МД, по изменению характеристик которых можно судить о изменении выбранного МП, в-третьих значительные объёмы исследуемых данных.

Именно поэтому, в настоящее время, наибольшие трудности представляет повышение точности прогнозирования и увеличение прогнозных сроков основных метеорологических параметров, что несомненно делает решение данной задачи актуальной.

Исходя из этого, в работе сформулирована **научная задача исследования:** разработка программного обеспечения для прогнозирования основных метеорологических параметров с использованием методов машинного обучения. В основе метода лежит выявление закономерностей, наблюдаемых МП и оценка их взаимовлияния по МД предшествующим оцениваемому периоду.

Целью исследования является методическое обеспечение оценки вероятности прогнозирования МП с использованием нейросетевых технологий.

Объектом исследования является методика построения вероятностного прогноза МП.

Предметом исследования служат системы и методы прогнозирования МП.

Для достижения цели необходимо решить следующие задачи:

- 1) провести анализ средств и методов реализации прогнозирования МП;
- 2) построить адекватную модель реализации прогнозирования МП;
- 3) создать программное средство для оценки МП на основе кластерного анализа;
- 4) осуществить тестирование программного средства прогнозирования МП.

Проводимые исследования основываются на положениях теории множеств, теории математической статистики, теории случайных процессов, а также на нейросетевых технологиях и методах объектно-ориентированного моделирования.

Основные положения, выносимые на защиту.

- 1) модель реализации прогноза изменения МП;
- 2) программное средство, для оценки модели прогноза изменения МП на основе кластерного анализа;
- 3) методика проведения тестирования модели и программного средства прогнозирования изменения МП.

Научная новизна работы заключается в реализации модели, для исследования МД методом нейросетевого анализа.

Основные научные результаты:

1. Предложена модель, позволяющая оценить вероятности изменения температуры.
2. Предложена методика комплексной оценки изменения температуры, на основе нейросетевого анализа.

Практическая значимость исследования заключается в следующем:

- 1) Использование моделирования по предложенной методике позволит повысить точность прогноза основных МП.
- 2) Комплексная оценка МД может использоваться для улучшения достоверности краткосрочного прогнозирования.

Значимость и достоверность результатов исследования подтверждается корректным использованием известного математического аппарата, а также проведением ряда экспериментов по анализу ретроспективных МД с

последующим сравнением результатов полученного прогноза изменения МП со значениями МП реального времени.

Структура и объем. Диссертация включает введение, пять глав, заключение, библиографический список и приложения.

Во введении обоснована актуальность исследований, сформулированы цель и задачи работы, определена практическая ценность и научная новизна выносимых на защиту результатов.

В первой главе анализируется современное состояние прогнозирования МП, рассматриваются модели построения прогноза и средства его реализации, по результатам анализа строится модель, позволяющая оценить вероятности изменения МП, отражающая структурные и функциональные зависимости между элементами системы на основе оценки главных компонент. Основное место в первой главе отводится анализу подходов к моделированию прогноза изменения МП.

Во второй главе осуществляется проектирование модели комплексной оценки МД, решающей задачу оптимизации и визуализации анализируемых данных, а также разрабатываются алгоритмы для проведения моделирования.

В третьей главе разрабатывается архитектура программного средства, реализующего кластерное распределение вероятностей изменения МП и разрабатывается методика проведения тестирования.

В четвёртой главе рассчитываются основные стоимостные характеристики реализации проекта и оцениваются сроки его окупаемости.

В пятой главе рассматриваются нормы предъявляемые к обеспечению безопасности труда при работе на персональном компьютере.

В заключении приводятся основные результаты исследований и делаются выводы по проделанной работе.

1. ИССЛЕДОВАТЕЛЬСКИЙ РАЗДЕЛ

1.1. Прогнозирование метеорологических прогнозов. Современное состояние

В настоящее время существуют два основных метода научного предсказания погоды: синоптический и численный (гидродинамический).

Синоптический метод базируется на построении и анализе синоптических карт (рис. 1.1), на которых выделяются отдельные объекты (циклоны, антициклоны, атмосферные фронты и т.д.), каждому из которых свойственны определённые типы погодных условий.

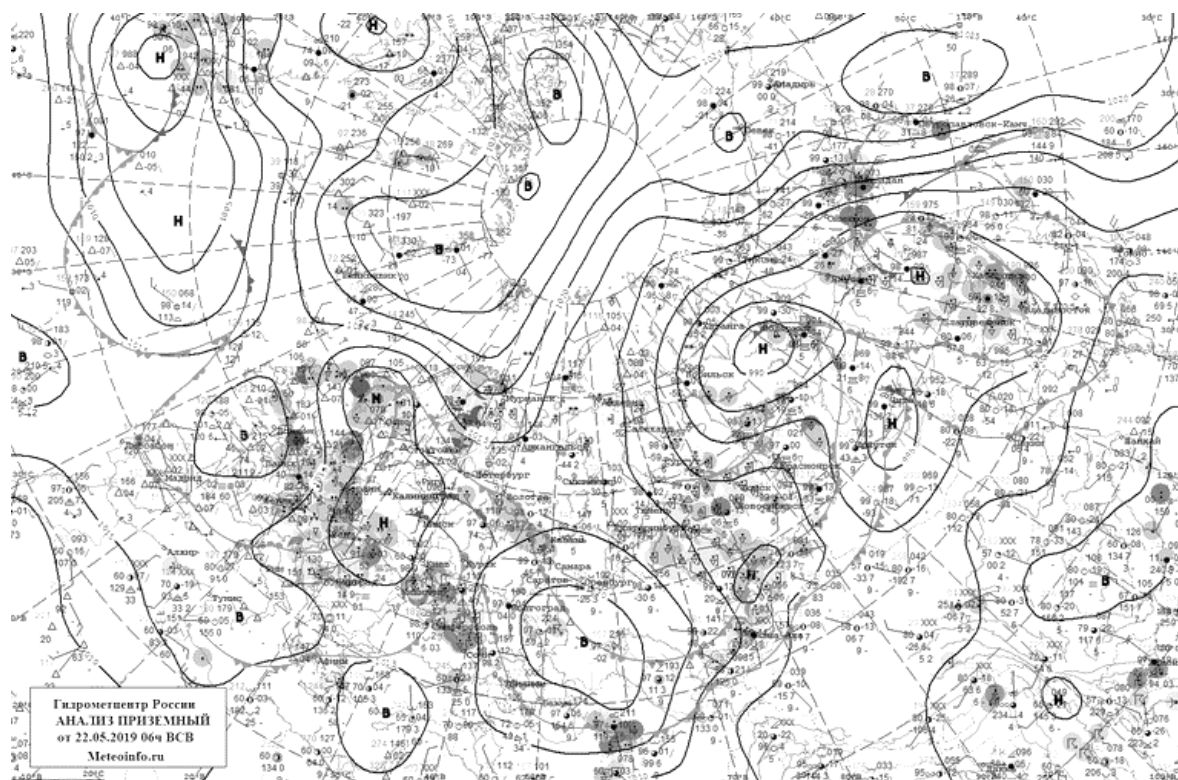


Рис 1.1. Синоптическая карта фактической погоды

Для построения карт выполняется сбор МД из различных источников (рис.1.2).

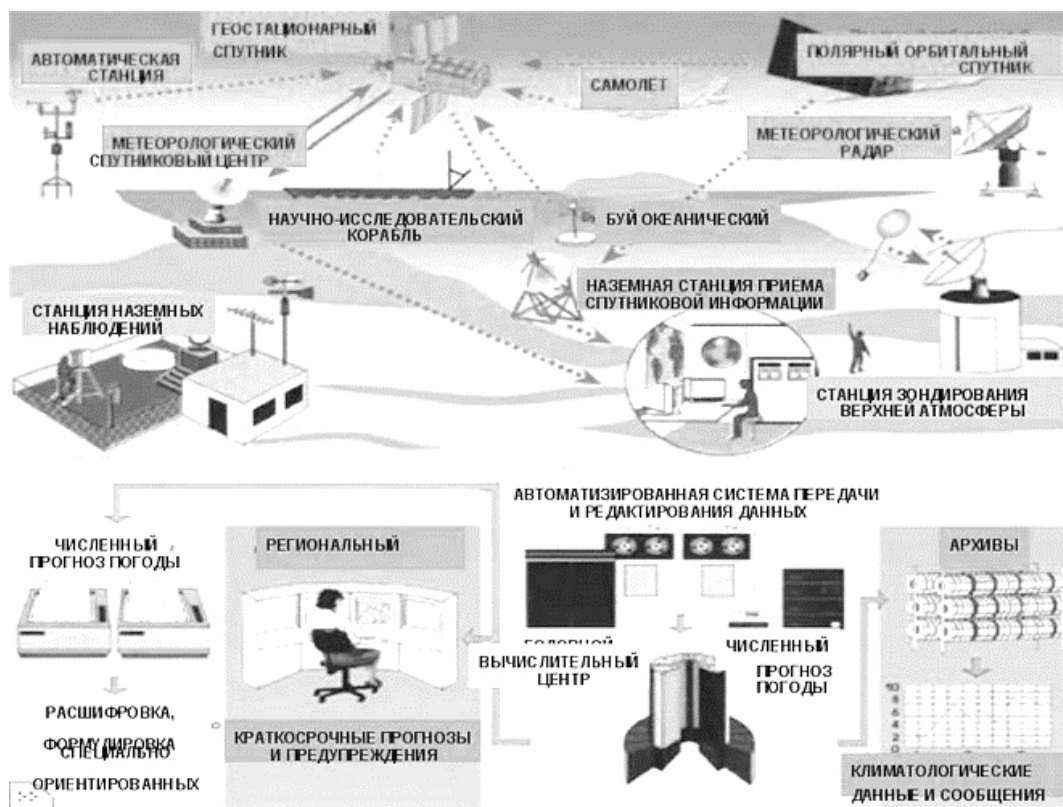


Рис 1.2. Источники метеоданных

Синоптические карты подразделяются на карты фактической погоды и прогностические. Синоптические карты предоставляют информацию о погоде, которая была в действительности на момент их выпуска, вторые – об погоде, которая ожидается в ближайшем времени. Карты обоих классов делятся на приземные и высотные (карты барической топографии).

Можно выделить характерные особенности и отличия подходов и методов оценки в различных статистических моделях построения прогноза с использованием оценки основных МП (табл. 1.1, рис. 1.4) [32].

Табл. 1.1. Сравнение методов прогнозирования

Модели прогнозирования	Достоинства	Недостатки
Регрессионные модели	Быстрота получения результата; Доступность промежуточных вычислений; Простота моделей; Разнородность решаемых задач.	Трудоемкость определения параметров; Возможность моделирования лишь линейных процессов; Сложность определения вида функциональной зависимости.
Авторегрессионные модели	Быстрота получения результата; Доступность промежуточных вычислений; Простота моделей; Разнородность решаемых задач.	Трудоемкость определения параметров; Возможность моделирования лишь линейных процессов.
Модели экспоненциального сглаживания	Простота моделей; Быстрота получения результата; Решение задач долгосрочного прогнозирования.	Отсутствие гибкости.
Нейросетевые модели	Возможность моделирования не линейных процессов; Адаптивность; Масштабируемость; Разнородность решаемых задач.	Сложность программной реализации; Отсутствие промежуточных вычислений; Высокие требования к непротиворечивости обучающей выборки.
Модели на базе цепей Маркова	Простота моделей;	Узкая применимость моделей. Невозможность решение задач прогнозирования с длинной памятью.

К классическим методам статистического прогнозирования можно отнести методы, построенные на фиксировании факта изменения того или иного метеорологического показателя, характерного для выбранного региона, с учётом его статистической значимости.

Реализация статистических методов прогнозирования невозможна без построения развитой сети станций наблюдения (наземного, морского и космического базирования) оборудованных многочисленными датчиками и аппаратурой анализа поступающих с них данных.

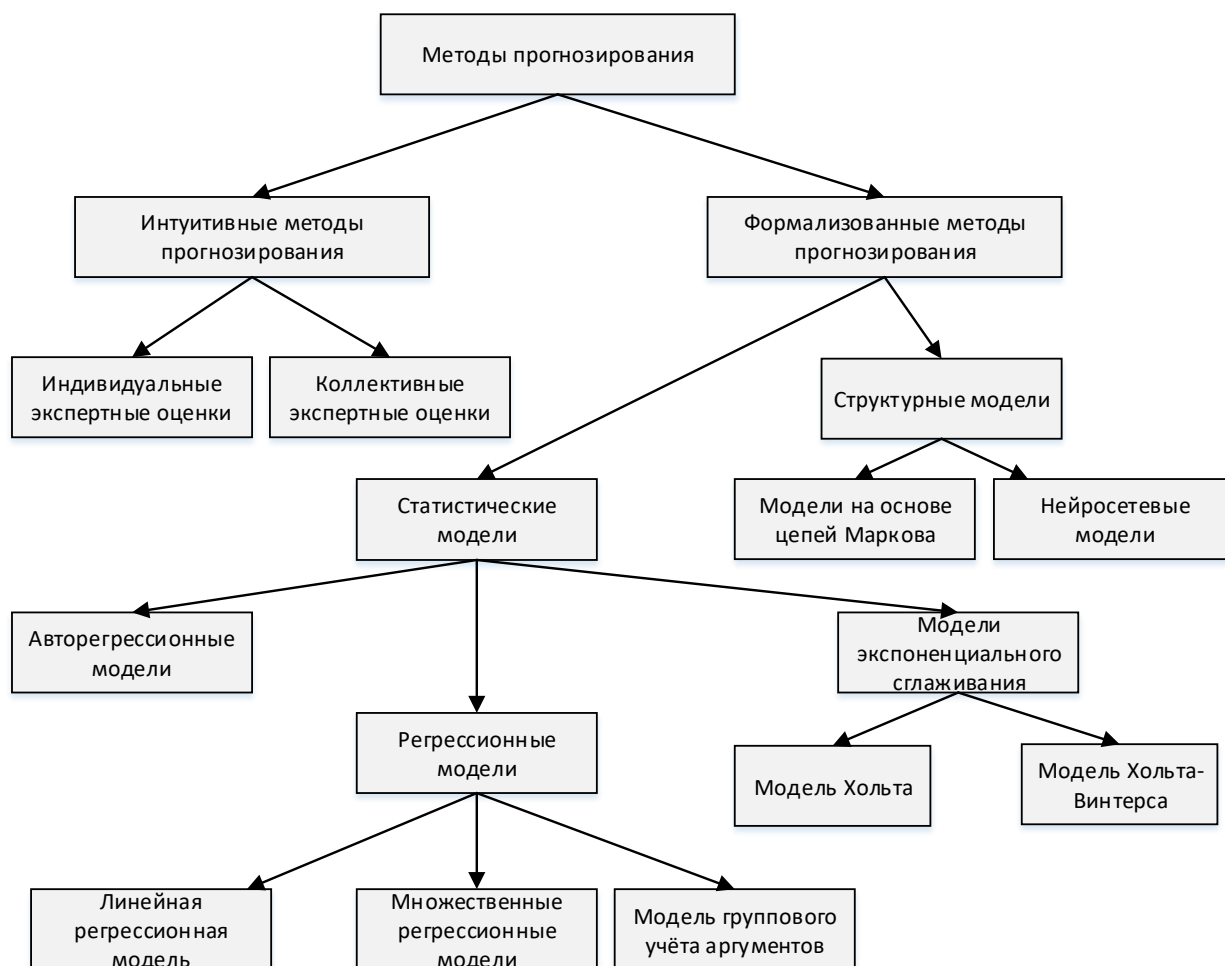


Рис. 1.4. Классификация методов и моделей прогнозирования

Необходимо так же отметить, что сегодня отсутствует научно обоснованная модель (методика), позволяющая реализовать даже точный среднесрочный прогноз (время, место, значение МП) какую бы теорию (строения Земли, вероятностей, математической статистики, ЦОС, ИИ и т.д.), с использованием какого-либо математического аппарата (регрессионного анализа, статистической оценки, вероятностного распределения и т.д.), основываясь на данных каких-либо наблюдениях (геолого-геофизических, геохимических, космических и т.п.), она бы не использовала.

Сложностями в реализации точного прогноза является дисперсия как самих МД, так и оценка их корреляции с МП, что проявляется как статистическая неоднозначность и в локализации места, и в оценке интенсивности изменения, и в определении времени прогнозируемого события.

Все рассмотренные методики построены на использовании очевидного факта, МД ретроспективного периода отличаются от МД реального периода,

необходимо только найти данные отличия, придать им весовую характеристику, построить пороговую шкалу и провести их совокупную оценку по критерию превышения установленного порога - реализацию прогноза. После реализации прогноза можно оценить его точность сравнивая с реально наблюдаемыми событиями. В том случае если совпадения происходят с некоторой вероятностью, превышающей статистическую ошибку, можно сделать вывод о его состоятельности. Для выявления отличий ретроспективного и реального периодов, можно использовать обширную базу многокритериальных данных, анализируемых, для выявления статистической значимости, ретроспективно.

Метеорологические данные, служащие основой прогнозирования изменения метеорологических параметров представлены на различных серверах и могут анализироваться практически в режиме онлайн.

Например, плотность расположения метеостанций на территории России, по данным [36], показана рис. 1.5.

В качестве примера сети различных датчиков и МД можно рассмотреть данные представляемые различными центрами метеорологии, например, Институтом глобального климата и экологии (ИГКЭ), Российская Федерация; Всероссийским научно-исследовательским институтом гидрометеорологической информации - Мировым центром данных (ВНИИГМИ-МЦД), Российская Федерация, а также данные различных каталогов МД, которые в описанных выше форматах располагаются в открытых базах данных.

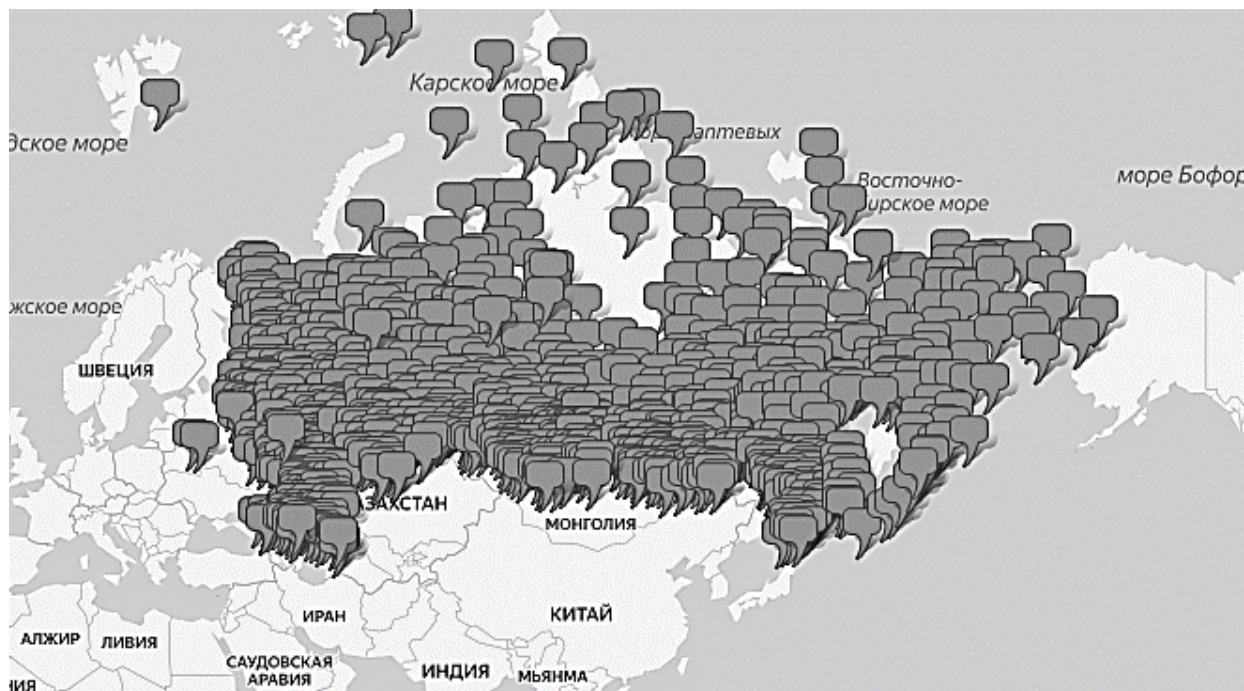
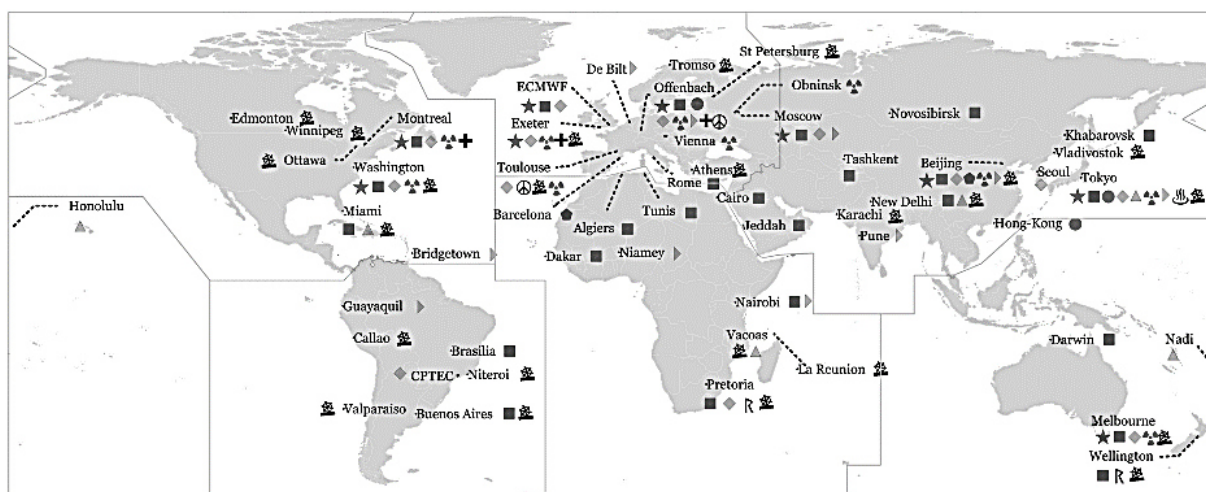


Рис. 1.5. Плотность расположения метеостанций на территории России

Расположение центров обработки МД в мире, по данным [18] показано на рис 1.7.



Legend

- ★ World Meteorological Centres* (9)
- ♦ Global Producing Centres for Long-Range Forecasts (13)
- + Global Producing Centres for Annual to Decadal Climate Prediction (3)
- RSMCs Geographic (25)
- RSMC Nowcasting (3)
- ▲ RSMCs TC (6)
- RSMCs Sand/Dust (2)
- ▴ Regional Climate Centres (11)
- ▲ RSMCs Nuclear Emergency Response** (10)
- ☪ RSMCs Non-Nuclear Emergency Response** (2)
- ☼ RSMCs Volcano Watch Services for International Air Navigation (1)
- ⚡ RSMCs Severe Weather Forecasting (2)
- ⚓ RSMCs Marine Meteorological Services (24)

* World Meteorological Centres are also Global Producing Centres for a) Deterministic Numerical Weather Prediction, b) Ensemble Numerical Weather Prediction, and c) Long-Range Forecasts.

** RSMC for nuclear and non-nuclear emergency response have Atmospheric Transport and Dispersion Modelling (ATDM) capabilities.

Рис. 1.7. Расположение центров обработки метеоданных

В табл. 1.2, показаны, находящиеся в открытом доступе, данные наблюдений, предоставляемых вышеперечисленными источниками.

Табл. 1.2. Данные наблюдений наземных станций

№ п.п.	Тип данных
1	Регистрация вариаций и абсолютных значений компонент магнитного поля Земли

2	Электромагнитное вертикальное зондирование ионосферы
3	Регистрация координат геодезического опорного пункта GPS, ГЛОНАСС
4	Метеорологические данные (скорость и направление ветра и воды их проводимость, температура, давление, влажность) и т.д.
5	Наклонное зондирование ионосферы по азимуту 20 градусов с помощью радара СуперДАРН
6	Вариации гравиметрического поля. Гравиметр установлен на пункт с измеренной абсолютной величиной поля.
7	Температура и влажность приповерхностного слоя почвы
8	Радиационный фон

В качестве примера сети различных датчиков, расположенных в акваториях можно рассмотреть данные представляемые системой Samos [37] обеспечивающей регулярный доступ к точным, высококачественным морским метеорологическим и приповерхностным океанографическим наблюдениям с исследовательских судов и судов добровольных наблюдений.

В качестве источников открытых геолого-геофизических данных могут также выступать: глобальные модели электрического и магнитного поля [38]; аномалии магнитного поля [39]; аномалии гравитационного поля [40]; данные тепловых потоков [41] и т.п. В табл. 1.3, показаны, находящиеся в открытом доступе, данные наблюдений системы Samos.

Табл. 1.3. Данные наблюдений станций морского базирования

№, п.п	Тип данных
1	Данные сонаров, эхолотов, доплеровских датчиков и гидрографических сонаров
4	Метеорологические данные (скорость и направление ветра и воды их проводимость, температура, давление, влажность) и т.д.
5	Данные глубинных зондов (проводимость воды, температура и т.д.)
6	Колебания земной коры, воды и воздуха - механические, оптические, электромагнитные и др.
7	Регистрация высокоточных координат геодезического опорного пункта - GPS/WAAS, ГЛОНАСС, Galileo и др.
8	Данные локального гравитационного поля Земли
9	Сила и направление локального магнитного поля Земли
10	Доля света, поглощенная или рассеянная частицами в воде или воздухе

Значения основных МП ретроспективно, в пограничном слое атмосферы от поверхности земли до 2 км, после чтения из массива МД форматов АЭРОСТАБ и АЭРОСТАС, можно представить в символьном формате. МП содержат статистики для высот: 0, 50, 100, далее с шагом 100 до 2000 м. Расчеты и дополнительный контроль данных в пограничном слое выполнен согласно работам [10, 11].

Файл данных для одной станции содержит временной ряд ежемесячных значений: средних, среднеквадратических отклонений и соответствующего числа наблюдений, использованных для их расчета. Формат записи для статистических характеристик представлен в табл. 1.4.

Табл. 1.4. Формат характеристик МП атмосферы

Призначная часть/ МП/число наблюдений	Обозначение	Число цифр	Единицы измерения	Константа отсутствия
Индекс станции	INDEX	5	-	нет
Год	GGGG	5	-	нет
Месяц	MM	3	-	нет
Срок	HH	3	GMT	нет
Стандартное значение высоты	Z	6	м	нет
Среднемесячные значения давления	MP	6	10·гПа	-9999
Среднеквадратические отклонения давления	SP	6	10·гПа	-9999
Число наблюдений для давления	NP	3	-	нет
Среднемесячные значения температуры	MT	6	10·°C	-9999
Среднеквадратические отклонения температуры	ST	6	10·°C	-9999
Число наблюдений для температуры	NT	3	-	нет
Среднемесячные значения дефицита точки росы	MD	6	10·°C	-9999
Среднеквадратические отклонения дефицита точки росы	SD	6	10·°C	-9999
Число наблюдений для дефицита точки росы	ND	3	-	нет
Среднемесячные значения скалярной скорости ветра	MS	6	10·м/с	-9999
Среднеквадратические отклонения скалярной скорости ветра	SS	6	10·м/с	-9999
Число наблюдений для скалярной скорости ветра	NS	3	-	нет
Среднемесячные значения зональной составляющей скорости ветра	MU	6	10·м/с	-9999
Среднеквадратические отклонения зональной составляющей скорости ветра	SU	6	10·м/с	-9999
Число наблюдений для зональной составляющей скорости ветра	NU	3	-	нет

Окончание табл. Формат характеристик МП атмосферы

Среднемесячные значения меридиональной составляющей скорости ветра	MV	6	10·м/с	-9999
Среднеквадратические отклонения меридиональной составляющей скорости ветра	SV	6	10·м/с	-9999
Число наблюдений для меридиональной составляющей скорости ветра	NV	3	-	нет

Пример записи основных МП, за 10 лет наблюдений, в форматах АЭРОСТАВ и АЭРОСТАС со станции наблюдения Москва ВДНХ

(Синоптический индекс - 27612 координаты 55.50 северной широты, 37.37 восточной долготы, высота над уровнем моря - 156 м), после обработки данных показан в таблице 1.5.

Табл. 1.5. МД по выбранному региону после обработки

INDEX 27612 (Москва-ВДНХ)												
Температура (MT)												
GG/MN	1	2	3	4	5	6	7	8	9	10	11	12
2008	-5,8	-1,5	1,8	9,4	11,3	15,6	19,1	17,5	10,9	8,9	2,3	-1,8
2009	-5,6	-5,4	-0,6	5,1	13,6	17,3	18,8	15,6	13,8	5,8	2,2	-6,5
2010	-14,5	-8,4	-1,1	8,3	16,7	18,8	26	21,8	11,7	3,8	2,7	-7,6
2011	-7,5	-11	-2	6,4	14,6	19	23,4	18,7	12,1	6,6	0,2	-0,1
2012	-6,8	-11,7	-3,1	8,2	15,1	17,1	20,9	17,7	12,9	6,5	1,6	-8,6
2013	-8,5	-3,5	-6,5	6,1	16,9	19,8	18,9	18,3	10,3	6,6	4	-1,7
2014	-8,6	-1,9	2,8	7	16	16,1	21,1	19,2	12,3	3,7	-1,3	-3,9
2015	-4,4	-2,2	2	6,1	14,2	17,9	18,3	17,6	13,8	4,4	0,8	0,2
2016	-10,1	-0,6	0,3	8,1	15	18,2	20,9	19,5	11,4	4,4	-2,7	-4,6
2017	-7,8	-4,6	2,4	5,3	10,9	14,4	17,9	18,8	13	5	0	0
Давление (MP)												
2008	1005	996,3	985,2	996,3	996,6	992,9	992,7	992,3	1000,6	997,7	995,3	1005,4
2009	995,1	997,7	994,2	998,6	995,1	997,3	993,9	996,8	999,2	996,4	998,3	998,8
2010	1008	1000,1	993	999,7	994,6	995	998,1	994,5	999,2	999,6	991,6	995,3
2011	998,5	998,6	997,7	994	999,1	993,7	994,5	996,6	992,6	998,3	1001	993,1
2012	1002	1005,3	991	991	997,8	993	996,7	995,1	995,5	995,2	999,3	1005
2013	994,8	999,6	992,7	998,4	996,4	996,5	992,4	998,3	990,7	998,6	994,2	994,6
2014	1003	1001,4	994,8	997,6	998,3	993,3	999,3	994,2	1001,8	1002,6	1010	995,3
2015	992,2	996,7	1005,9	989,2	993,4	995,8	991	1001,3	1001,1	1002,1	993,2	995,6
2016	994,3	995,1	995,7	992,4	996,5	995,5	993	998,2	995,4	1009,1	997,9	994,5
2017	998,3	997,5	995,3	993,5	995,5	989,1	991,2	998,8	1001,5	992,7	997,8	992,6

МД (показаны частично) реального периода (пять суток) для указанной станции, можно получить с различных погодных сервисов (табл. 1.6, рис. 1.5) или напрямую со станции наблюдения в формате КН-03.

Табл. 1.6. МД реального периода

Дата	Макс. t, °C	Мин. t, °C	Ср. t, °C	Эфф. t, °C	Атм. давление, ГПа	Ск. ветра, м/с	Влаж- ть, %	Нап-е ветра	Точки росы, t °C	Абс. влаж-ть, г/м³
20 мая	19.3	8.1	14.1	13.2	996.6	1	76,8	СЗ	6.4	5.12
21 мая	19.1	8.2	13.9	13.1	996.4	1	77	СЗ	6.4	5.13
22 мая	19.0	8.6	14.1	13.4	995.8	1	76,4	ЮЗ	6.4	5.12
23 мая	18.6	8.4	13.8	13.2	995.7	1	76,3	СЗ	6.4	5.12
24 мая	19.0	8.3	13.9	13.3	995.6	1	76,5	ЮЗ	6.4	5.12
25 мая	20.0	8.4	14.4	13.7	995.7	1	77,1	СЗ	6.4	5.13

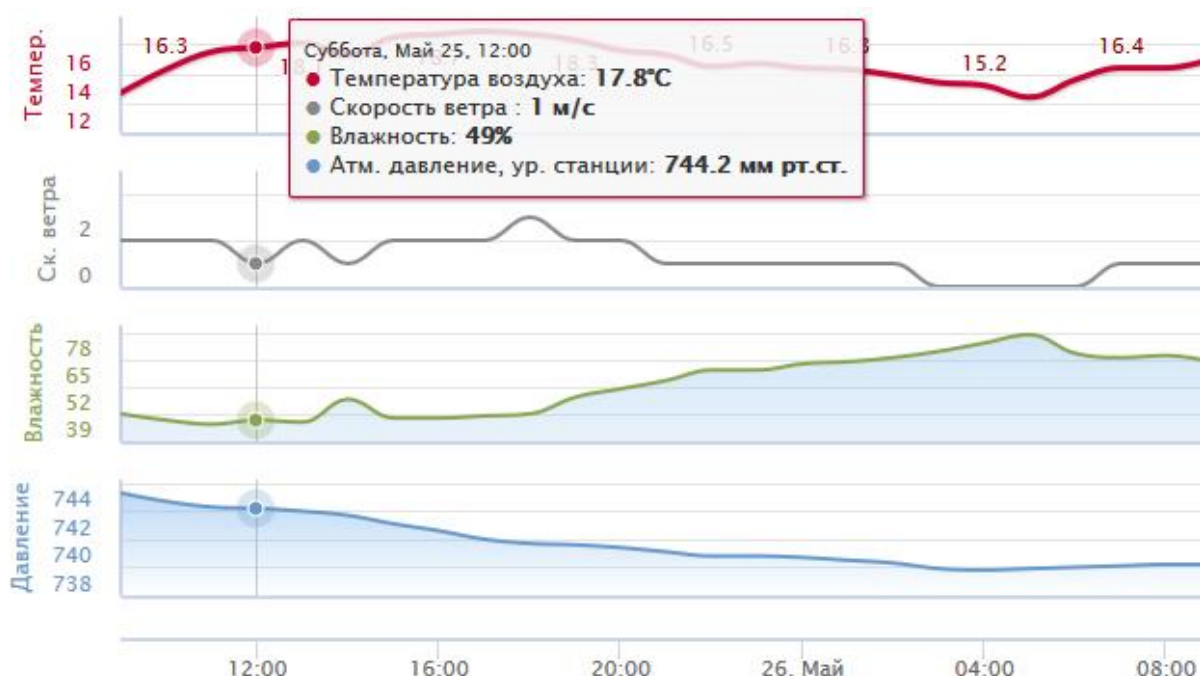


Рис. 1.6. МД, за сутки, со станции метеонаблюдения «Москва ВДНХ»

Помимо этого, для анализа МП предшествующих данному событию, можно оценить состояние дополнительных МП, основываясь на данных открытых источников, описанных выше.

1.2. Анализ средств и методов автоматизации краткосрочного прогнозирования

В зависимости от решаемых задач, множество программного обеспечения, так или иначе связанного с обработкой и анализом МД, можно классифицировать по типу решаемых задач на мониторы МП, утилиты для конвертации и чтения записи МД, обработчики МД (анализаторы, программы визуализации) и собственно программы для построения прогнозов МП.

Рассмотрим некоторые бесплатные программы анализаторы.

Delivery - программа анализа, в том числе и МД, на основе Бейсовского метода инверсии, позволяющая осуществлять анализ основных МП, по методике оценки кинематики распространения волн различной этимологии [42].

Пакет обработки, анализа и визуализации (2-D и 3-D) геофизических данных JRG. Прогноз построен на анализе всплесков МП с их предварительной фильтрацией и ранжированием. В основе анализа лежит нейросетевой метод поиска главных компонент с динамическим определением порога выборки [43].

Пакет моделирования и анализа МП методом томографической инверсии TEMIMAGE позволяем смоделировать распространение и участки аномальных отклонений МП. Анализ полученных данных осуществляется методом сравнения с данными ретроспективного периода, при помощи регрессионного анализа и нахождением интегральной зависимости выбранных МП от исследуемых МД [44].

Для анализа (оценка неопределённости, уровня изменений, значения взаимосвязи) геофизических, в основном приповерхностных данных на наличие аномалий можно использовать программу WaveletExtractor осуществляющую подготовку исходных данных методом вейвлет преобразования с последующим сравнением с эталонным методом спектрального анализа [45].

Использование спектрального анализа (изменение частоты дискретизации, преобразование Фурье, свёртка) с последующей статистической обработкой полученных результатов (интерполяция, слияние, тренд и т.д.) с использованием заранее подготовленных данных можно выполнить в проекте МЕТЕО. Данный программный продукт не осуществляет подготовку данных и не выполняет самого прогнозирования МП [45].

Для выявления главных компонент различных МД, полученных от метеостанций, в различных форматах, можно использовать программу GeoSeis, выявление отличий от ретроспективного периода построено на нейронной сети обратного распространения [46].

Рассмотрим инновационные программные продукты, позволяющие выполнять прогнозирование МП на основе методик, рассмотренных в п.1.1.

На кафедре физики атмосферы Московского государственного университета осуществляется попытка прогнозирования изменения МП с применением кластерного анализа радиотомографических данных ионосферы наблюдаемых в реальном времени МД [48]. Результат прогноза формируется на основе оценки распространения изменения основных МП, оценки радиационного фона, геофизических, геохимических и геодезических данных полученных от станций наблюдения расположенных в узловых точках, расстояние между которыми фиксировано и не превышает нескольких сотен

километров. Каждая узловая точка реализует комплексную оценку вероятности изменения МП, по совокупности которых строится окончательный прогноз. Кластерный анализ реализуется при помощи нейросети SOAM (самоорганизующейся карты Кохонена).

Основываясь на предположение о возможности восстановлении зависимостей МП по данным прошлых лет В.Н. Вапником предложен алгоритм идентификации периодов МП с помощью разделения пространства оцениваемых признаков - кластеризации. Для проведения натурных экспериментов реализован программно-алгоритмический комплекс строящий комплексную оценку МД, используя предварительно подготовленные, содержащие весовые характеристики, данные МП. Оценка вероятности и кластеризация осуществляется при помощи карты SOM. Результаты экспериментов подтверждают высокую достоверность прогноза, с учётом тщательного отбора и анализа ретроспективных МД [49].

В «Яндекс.Погоде» используется технология Метеум, основанная на метеорологических моделях и машинном обучении. Построенная на базе Матрикснета и обученная на больших массивах данных, нейронная сеть MLP, после предварительного анализа главных компонент МД позволяет повысить точность прогноза [52].

Согласно проведённому анализу, учитывая рассмотренные методики и модели на основе которых они реализованы, принимая во внимание классификацию программного обеспечения, а также с учётом выводов, полученных выше, можно построить обобщённый алгоритм прогнозирования метеорологических параметров (рис. 1.6).

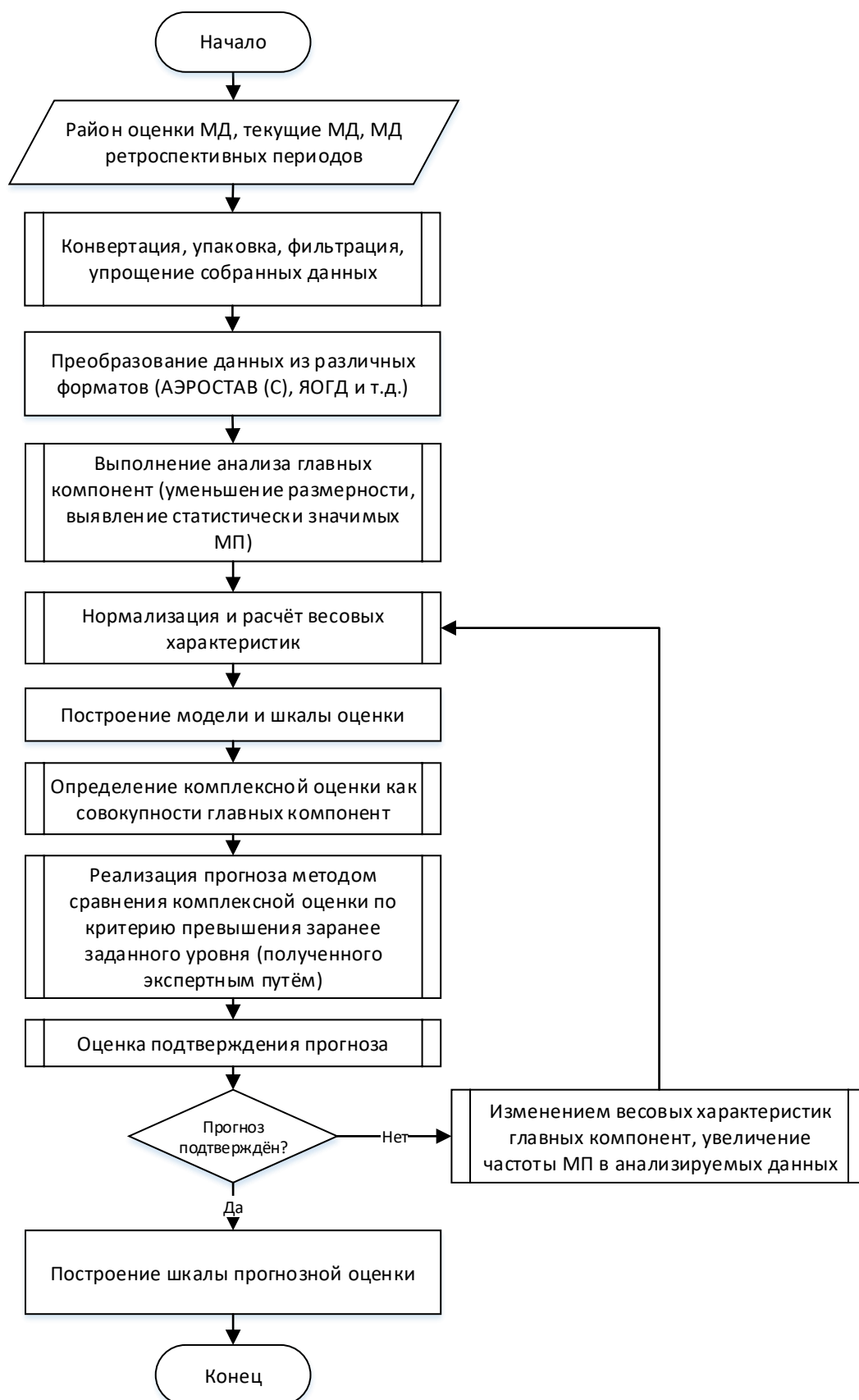


Рис. 1.6. Модель прогнозирования метеорологических параметров

Анализ программного обеспечения, в части прогнозирования метеорологических параметров, на основе нейросетевого анализа, позволяет сделать следующий вывод: все рассмотренные программы оперируют данными

от метеорологических станций (наземного, воздушного и космического базирования); все рассмотренные программы используют определённую модель оценки построенную на отличиях метеорологических данных исследуемых периодов от ретроспективных; все рассмотренные программы построены с использованием того или иного математического аппарата, позволяющего реализовать анализ отличий сравниваемых периодов с использованием синтетической градации весов того или иного параметра или их совокупности; в рассмотренных программах прогнозная оценка строится методом сравнения рассчитанной величины или комплексной оценки возможного события, по критерию превышения экспериментально установленного порога или при помощи одной из рассмотренных математических прогностических моделей.

Исходя из этого при реализации программы прогнозирования изменений основных МП будет использован алгоритм, показанный на рис. 1.6.

1.3. Модель подготовки данных для прогнозирования метеорологических параметров

Учитывая проанализированные статистические модели построения прогноза, используемые ими методы, а так же реализованный алгоритм (рис. 1.6), рассмотрим системы самоорганизации (без учителя), основанные на принципе обучения Хебба [54] и опишем метод анализа главных компонент* (далее АК), который будет использоваться для уменьшения размерности и получения репрезентативной выборки данных в статистической системе обработки МД.

Обучение на основе самоорганизации заключается в последовательном изменении синаптических весов нейронной сети в ответ на возбуждающие сигналы, производимые в соответствии с заранее определенными правилами [55-58].

Для того чтобы сеть достигла самоорганизации, обратная связь между изменениями в синаптических весах и изменениями в активности сигналов должна быть положительной. В соответствии с этим можно сформулировать основные принципы самоорганизации [59].

* Анализ (метод) главных компонент (Principal Component Analysis - PCA) метод многомерного анализа [517], [857].

ПРИНЦИП 1. Изменение синаптических весов ведет к самоусилению сети. Используемый для предварительного анализа предвестников механизм на самом деле является подтверждением постулата обучения Хебба [59].

ПРИНЦИП 2. Ограниченность ресурсов ведет к конкуренции между синапсами и, таким образом, к выбору наиболее успешно развивающихся синапсов за счет других (т.е. наиболее подходящих) [59].

ПРИНЦИП 3. Модификация синаптических весов ведет к кооперации. Присутствие сильного синапса может усилить и другие синапсы в свете общей системы конкуренции в сети.

ПРИНЦИП 4. Порядок и структура сигналов активации содержат избыточную информацию, накапливаемую сетью в форме знаний, являющихся необходимым условием самоорганизующегося обучения.

Некоторая часть знаний может быть получена с помощью наблюдения за статистическими параметрами, такими как среднее значение, дисперсия и матрица корреляции входных данных.

Рассмотренные принципы самоорганизующегося обучения образуют базис для реализации адаптивных алгоритмов АК, а также для самоорганизующихся карт Кохонена.

Метод АК интенсивно используется для сокращения размерности описания, выделения значимой информации, визуализации данных и др. Ещё один вариант использования - это анализ соответствий, другие области применения это техника сокращения динамических моделей которая интенсивно применяется в климатологии [64, 67].

С точки зрения задачи статистического прогнозирования МП практическое значение АК состоит в том, что он обеспечивает эффективный способ сокращения размерности [67, 72, 81]. В частности, можно сократить количество признаков, необходимых для эффективного представления данных, либо, сократить данные одиночного датчика МП, например, термометра, устраняя те линейные комбинации в (1.27), которые имеют малые дисперсии, и оставляя те, дисперсии которых велики. Пусть $\lambda_1, \lambda_2, \dots, \lambda_l$ - наибольшие l собственных

значений матрицы корреляции R . Тогда вектор данных x можно аппроксимировать, отсекая члены разложения (27) после l -го слагаемого:

$$\hat{x} = \sum_{j=1}^l a_j q_j = [q_1, q_2, \dots, q_n] \begin{bmatrix} a_1 \\ a_2 \\ \dots \\ a_n \end{bmatrix}, l \leq m. \quad (1.28)$$

Имея исходный вектор x , с помощью выражения (1.25) можно вычислить главные компоненты из (1.28) следующим образом:

$$\begin{bmatrix} a_1 \\ a_2 \\ \dots \\ a_l \end{bmatrix} = \begin{bmatrix} q_1^T \\ q_2^T \\ \dots \\ q_l^T \end{bmatrix} x, \quad l \leq m. \quad (1.29)$$

Линейная проекция (1.29) из $\mathcal{R}^m$ в $\mathcal{R}^l$ (т.е. отображение из пространства данных в пространство признаков) представляет собой метод кодировки для приближенного представления вектора данных x (рис. 1.7, а). Соответственно линейная проекция (1.28) из $\mathcal{R}^l$ в $\mathcal{R}^m$ (т.е. обратное отображение пространства признаков в пространство данных) представляет собой метод декодирования (см. рис. 1.7, б).

Необходимо отметить, что наиболее значимые собственные значения $\lambda_1, \lambda_2, \dots, \lambda_i$ не участвуют в вычислениях (1.28) и (1.29), но необходимы для определения количества АК, используемых для кодирования и декодирования.

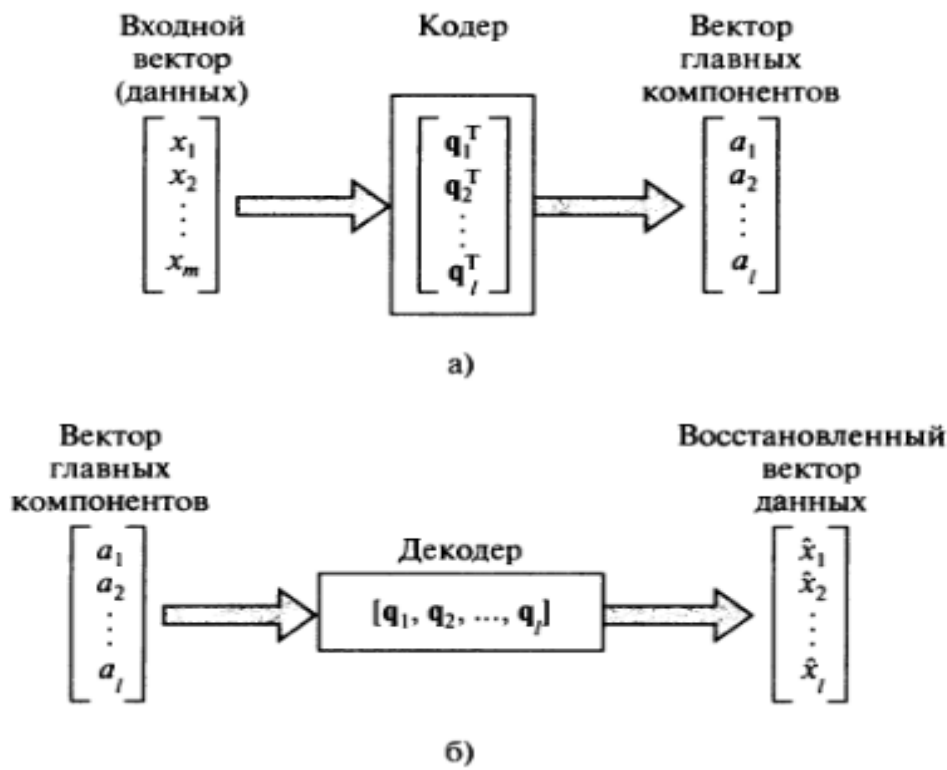


Рис. 1.7. Два этапа АК: кодирование (а) и декодирование (б)

Вектор ошибки аппроксимации e равен разности между вектором исходных данных x и вектором приближенных данных $\hat{x}$ (рис. 1.8):

$$e = x - \hat{x}. \quad (1.30)$$

Подставляя (1.27) и (1.28) в (1.30), получим:

$$e = \sum_{j=l+1}^m a_j q_j. \quad (1.31)$$

Вектор ошибки e является ортогональным вектору приближенных данных x (рис. 1.6). Другими словами, скалярное произведение векторов e и $\hat{x}$ равно нулю. Используя (28) и (31), это свойство можно доказать:

$$e^T \hat{x} = \sum_{i=l+1}^m a_i q_i^T \sum_{j=1}^l a_j q_j = \sum_{i=l+1}^m \sum_{j=1}^l a_i a_j q_i^T q_j = 0, \quad (1.32)$$

где учитывается второе условие выражения (1.19). Соотношение (1.32) называют принципом ортогональности.

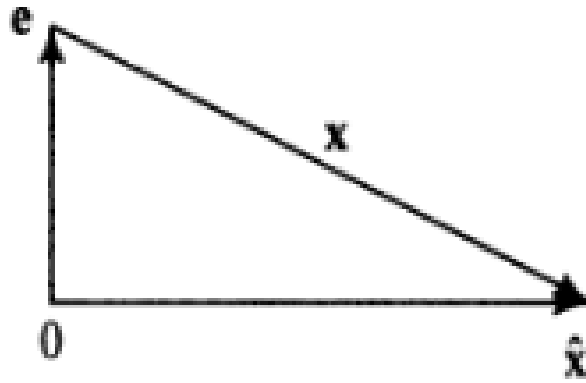


Рис. 1.8. Взаимосвязь вектора x , его реконструированной версии $\hat{x}$ и вектора ошибки e

Общая дисперсия m компонентов вектора данных x составляет (согласно (1.7) и (1.22)):

$$\sum_{j=1}^m \sigma_j^2 = \sum_{j=1}^m \lambda_j, \quad (1.33)$$

где σ_j^2 - дисперсия j -го главного компонента a_j . Общая дисперсия l элементов приближенного вектора $\hat{x}$ равна:

$$\sum_{j=1}^l \sigma_j^2 = \sum_{j=1}^l \lambda_j. \quad (1.34)$$

Таким образом, общая дисперсия $(l - m)$ элементов вектора ошибки аппроксимации $x - \hat{x}$ равна:

$$\sum_{j=l+1}^m \sigma_j^2 = \sum_{j=l+1}^m \lambda_j. \quad (1.35)$$

Числа $\lambda_{l+1}, \dots, \lambda_m$ являются наименьшими $(m - l)$ собственными значениями матрицы корреляции R . Они соответствуют слагаемым, исключенным из разложения (1.28), используемого для построения приближенного вектора $\hat{x}$. Чем ближе эти собственные значения к нулю, тем более эффективным будет сокращение размерности (как результат применения АК вектора данных x) в представлении информации исходных данных. Таким образом, для того чтобы обеспечить сокращение размерности входных данных, нужно вычислить собственные значения и собственные векторы матрицы корреляции векторов входных данных, а затем ортогонально проецировать эти данные на подпространство, задаваемое собственными векторами,

соответствующими доминирующим собственным значениям этой матрицы. Этот метод представления данных обычно называют пространственной декомпозицией [72].

Анализ МД производится с многомерными множествами данных, в которых кластерная структура данных скрыта, именно поэтому необходимо выполнять статистический анализ, аналогичный описанному выше анализу главных компонент [73].

2. СПЕЦИАЛЬНЫЙ РАЗДЕЛ

2.1. Обработка многомерного сигнала фильтром Хебба

Существует тесная взаимосвязь между поведением самоорганизующихся нейронных сетей и статистическим методом АК.

Один линейный нейрон с Хеббовским правилом адаптации синаптических весов может быть преобразован в фильтр для выделения первого главного компонента входного распределения [75].

Рассмотрим нейронную модель выход которой является линейной комбинацией входных векторов, показанную на рис. 2.1, а. Нейрон получает множество из m входных сигналов $x_1, x_2, \dots, x_m$ через соответствующее множество m синапсов $w_1, w_2, \dots, w_m$. Выход полученной модели можно определить, как

$$y = \sum_{i=1}^m w_i x_i. \quad (2.1)$$

Согласно постулату Хебба, синаптический вес w_i изменяется во времени, сильно возрастая, если предсинаптический сигнал x_i и постсинаптический сигнал y совпадают друг с другом.

Общая работа нейрона представляется как комбинация двух графов передачи сигнала, показанных на рис. 2.5. Граф на рис. 2.5, а отражает зависимость выхода $y(n)$ от синаптических весов $w_1(n), w_2(n), \dots, w_m(n)$, согласно (2.1). Графы передачи сигнала на рис. 2.1, б иллюстрируют выражения (2.6) и (2.7). Передаточная функция z^{-1} в средней части графа представляет собой оператор единичной задержки. Выходной сигнал $y(n)$ на рис. 2.1, а выступает в роли передаточной функции на рис. 2.1, б. Граф на рис. 2.1, б ясно показывает следующие две формы внутренней обратной связи, действующей в нейроне:

- положительная обратная связь для самоусиления и роста синаптических весов $w_i(n)$ в соответствии с внешним входным сигналом $x_i(n)$;
- отрицательная обратная связь $-y(n)$ для контроля роста связей, стабилизирующая синаптический вес $w_i(n)$.

Слагаемое-произведение $(-y(n)w_i(n))$ связано с фактором забывания, который часто используется в правилах обучения, но с одним отличием: фактор забывания становится более явно выраженным с усилением выходного сигнала $y(n)$. Такой тип управления имеет чёткое нейробиологическое объяснение [75].

Приведём матричную формулировку алгоритма.

Введём следующие обозначения:

$$x(n) = [x_1(n), x_2(n), \dots, x_m(n)]^T \quad (2.8)$$

и

$$w(n) = [w_1(n), w_2(n), \dots, w_m(n)]^T. \quad (2.9)$$

Входной вектор $x(n)$ и вектор синаптических весов $w(n)$ обычно являются реализациями случайных векторов. Используя это векторное представление, выражение (2.1) можно переписать в форме скалярного произведения:

$$y(n) = x^T(n)w(n) = w^T(n)x(n) \quad (2.10)$$

Аналогично, выражение (2.5) можно переписать в следующем виде:

$$w(n+1) = w(n) + \eta y(n)[x(n) - y(n)w(n)]. \quad (2.11)$$

Подставляя (2.10) в (2.11), получим:

$$w(n+1) = w(n) + \eta [x(n)x^T(n)w(n) - w^T(n)x(n)x^T(n)w(n)w(n)]. \quad (2.12)$$

Алгоритм обучения (2.12) представляет собой нелинейное стохастическое разностное уравнение (алгоритм самоорганизующегося обучения), которое позволяет провести анализ сходимости на основе общих методов анализа сходимости стохастических алгоритмов аппроксимации. Уравнение (2.13) являясь частным случаем алгоритма стохастической аппроксимации:

$$w(n+1) = w(n) + \eta(n)h(w(n), x(n)), \quad n = 1, 2, \dots \quad (2.13)$$

где $\eta(n)$ состоит из положительных скалярных величин, а $h(.,.)$ является детерминированной функцией с некоторым условием регулярности. Данная

функция определяет полную структуру алгоритма в матричной форме. В [46] и [47] приведены разные подходы к анализу сходимости, приводящие эту задачу к задаче сходимости дифференциальных уравнений (ODE).

С помощью анализа устойчивости ODE можно исследовать сходимость рекурсивного алгоритма (2.13), определяющего фильтр для извлечения максимального собственного значения.

Из (2.13) видно, что функция коррекции $h(w, x)$ определена следующим образом:

$$h(w, x) = x(n)y(n) - y^2(n)w(n) = x(n)x^T(n)w(n) - [w^T(n)x(n)x^T(n)w(n)] w(n). \quad (2.14)$$

Рассмотрим пример согласованного фильтра, необходимого для предварительной обработки векторов содержащих МД, где случайный вектор $X(n)$, составленный как: $X(n) = s + V(n)$, где s - фиксированный единичный вектор, представляющий компонент полезного сигнала при анализе МП; $V(n)$ - компонент шума со средним значением 0.

Матрица корреляции такого входного вектора имеет следующий вид: $R = E[X(n)X^T(n)] = ss^T + \sigma^2 I$, где σ^2 - дисперсия элементов вектора шума $V(n)$; I - единичная матрица. Наибольшее собственное значение такой матрицы будет следующим: $\lambda_1 = 1 + \sigma^2$.

Соответствующий собственный вектор q_1 имеет вид $q_1 = s$.

Не составляет труда показать, что это решение удовлетворяет уравнению для собственных значений: $Rq_1 = \lambda q_1$.

Для описанной в данном примере ситуации самоорганизующийся линейный нейрон (при его сходимости к устойчивому состоянию) выступает в качестве согласованного фильтра, его импульсный отклик (представленный синаптическими весами) соответствует компоненту полезного сигнала s входного вектора $X(n)$.

Для эвристического понимания того, как обобщенный алгоритм Хебба работает на самом деле, в первую очередь запишем версию алгоритма (2.18) в матричном представлении:

$$\Delta w_{ji}(n) = \eta y_j(n) x'_i(n) - \eta y_j^2(n) w_{ji}(n), \quad j = 1, 2, \dots, l, \quad (2.24)$$

где

$$x'(n) = x(n) - \sum_{k=1}^{j-1} w_k(n) y_k(n). \quad (2.25)$$

Вектор $x'(n)$ представляет собой модифицированную форму входного вектора. Основываясь на представлении (2.24), можно сделать следующие наблюдения [87].

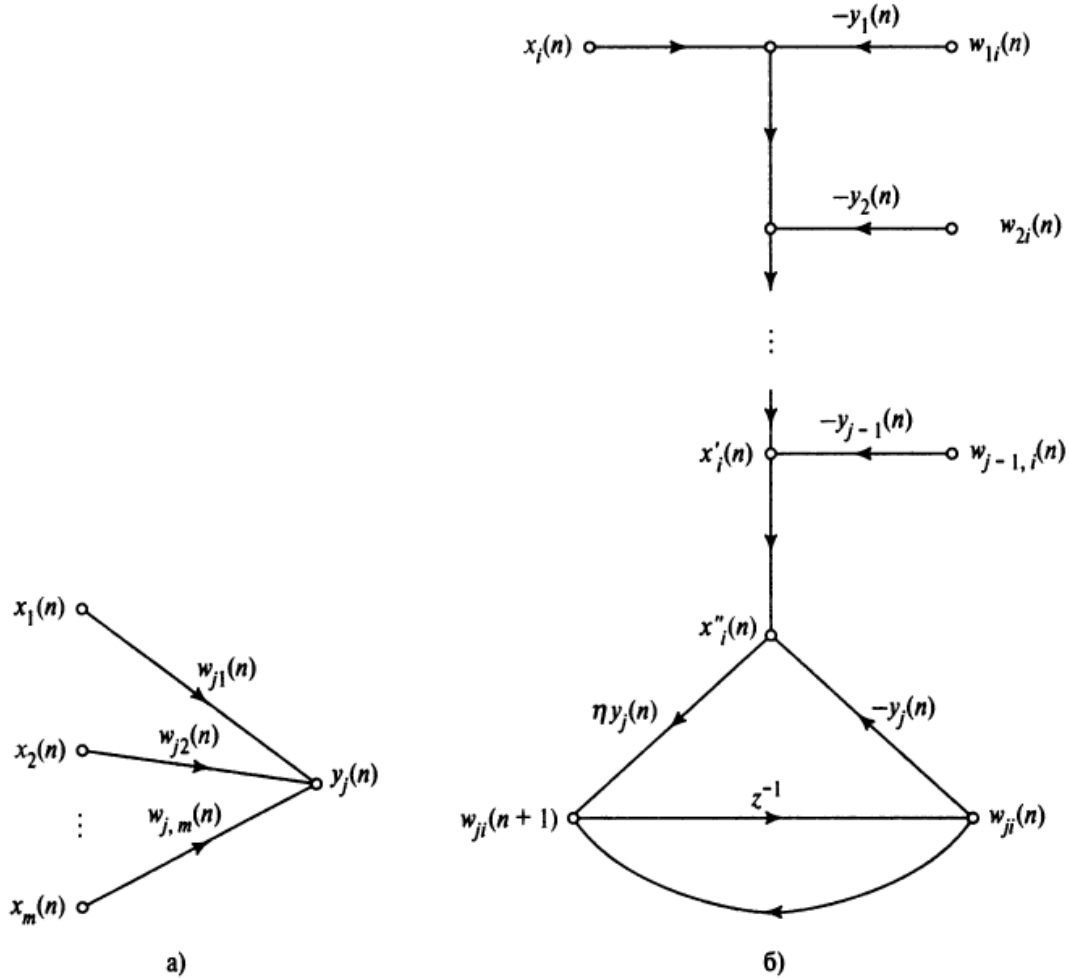


Рис. 2.3. Представление обобщенного алгоритма Хебба в виде графа передачи сигнала: граф уравнения (2.16) (а); граф выражений (2.17), (2.18) (б)

Для первого нейрона сети прямого распространения (см. рис. 2.2): $j = 1$: $x'(n) = x(n)$.

Для этого случая обобщенный алгоритм Хебба сводится к виду (2.11), записанному для одиночного нейрона. Выше показано, что этот нейрон извлекает первый основной компонент входного вектора $x(n)$.

1. Для второго нейрона сети на рис. 2.3 можно записать: $j = 2: x'(n) = x(n) - w_1(n)y_1(n)$.

2. Для третьего нейрона можно записать: $j = 3: x'(n) = x(n) - w_1(n)y_1(n) - w_2(n)y_2(n)$

3. Продолжая эту процедуру для оставшихся нейронов сети прямого распространения (см. рис. 2.3), получим, что каждый из выходов сети, обученный с помощью обобщенного алгоритма Хебба (2.18), представляет собой отклик на конкретный собственный вектор матрицы корреляции входного вектора, причем отдельные выходы упорядочены по убыванию ее собственных значений.

Этот метод вычисления собственных векторов аналогичен методу, получившему название процесса исчерпания [89]. Он использует процедуру, аналогичную ортогонализации Грама-Шмидта [102].

2.2. Вычислительный алгоритм предварительной подготовки данных

Вычисления, выполняемые рассмотренным алгоритмом можно записать в следующем виде:

Вычисления, выполняемые обобщенным алгоритмом Хебба, являются простыми, и их можно описать следующей последовательностью действий.

1. В момент времени $n = 1$ инициализируем синаптические веса w_{ji} сети случайными малыми значениями. Назначаем параметру скорости обучения η некоторое малое положительное значение.

2. Для $n = 1, j = 1, 2, \dots, l$ и $i = 1, 2, \dots, m$ вычислим:

$$y_j(n) = \sum_{i=1}^m w_{ji}(n)x_i(n),$$

$$\Delta w_{ji}(n) = \eta [y_j(n)x_i(n) - y_j(n) \sum_{k=1}^j w_{ji}(n)y_k(n)],$$

где $x_i(n)$ - i -й компонент входного вектора $x(n)$ размерности $m \times 1$; l – требуемое число главных компонентов.

3. Увеличиваем значение n на единицу, переходим к шагу 2 и продолжаем до тех пор, пока синаптические веса w_{ji} не достигнут своих установившихся значений. Для больших n синаптические веса w_{ji} нейрона j сходятся к i -му

компоненту собственного вектора, связанного с j -м собственным значением матрицы корреляции входного вектора $x(n)$.

В [90] указывается на значимость множества алгоритмов, предложенных в [84], [81], для АК (т.е. для алгоритмов Хебба, рассмотренных выше), в алгоритме иерархической кластеризации. Они продвигают гипотезу о том, что иерархическая кластеризация может выступать в качестве фундаментального свойства (по крайней мере частично) памяти, которое можно использовать для распознавания свойств окружающей среды. Отсюда следует вывод, что самоорганизующийся анализ главных компонент может играть существенную роль в иерархической кластеризации обучаемых областей не из-за его свойств оптимальности восстановления, а благодаря его встроенному свойству создавать проекции кластеров с хорошим разделением.

Рассмотренный алгоритм АК будет использован в решении задачи квантования вектора обучения (восстановления, предварительной подготовки и сжатия МД), полученных из различных источников, представляющих множество МП.

2.3. Самоорганизующиеся системы и обучение сети

Архитектура самоорганизующихся систем может принимать множество совершенно различных форм. Например, такая сеть может состоять из входного и выходного слоя, связанных прямыми связями, и включать косвенные связи между нейронами второго слоя. Еще одним примером могут служить многослойные сети прямого распространения, в которых самоорганизация проявляется при переходе от одного слоя к другому. В обоих случаях процесс обучения состоит в периодически повторяющемся изменении синаптических весов всех связей в системе в ответ на подачу входных сигналов в соответствии с предписанными правилами до получения конечной конфигурации системы.

Важным свойством нейронных сетей является их способность обучаться на основе примеров из окружающей среды и с помощью этого обучения повышать производительность работы. Целью обучения является построение желаемого отображения входного сигнала в выходной, которое сеть должна аппроксимировать. Целью алгоритмов самоорганизующегося обучения является

выявление в множестве входных данных существенных образов или признаков, причем этот процесс проходит без участия учителя. Для этого алгоритм реализует множество правил локальной природы, что позволяет обучаться вычислению отображения входного сигнала на выходной с требуемыми свойствами. Здесь термин «локальный» подразумевает следующее. Изменения синаптических весов нейрона определяются только непосредственными соседями этого нейрона. Модели сетей, обучаемые на основе принципа самоорганизации, в гораздо большей мере отражают свойства природных явлений, нежели архитектуры, обучаемые с учителем.

Как было отмечено, для обучения необходима избыточность входных данных, ярко проявляющаяся в наличии большого количества МП. Выше был описан алгоритм, позволяющий осуществить получение статистически значимых данных на этапе подготовки. Опишем собственно анализ данных присущих самообучающейся нейронной сети. В качестве модели анализа данных рассмотрим алгоритм векторного кодирования – модель Кохонена [73]. Эта модель реализует топологическое отображение (см. рис. 2.4), которое оптимально размещает фиксированное количество векторов (т.е. кодовых слов) во входное пространство более высокой размерности и, таким образом, облегчает анализ данных.

Исходя из этого, модель Кохонена можно представить, как подход векторного квантования, включающий кодирование и декодирование. Этот подход основывается на положениях теории коммуникаций [73], [74].



Рис. 2.4. Самоорганизующаяся карта признаков в модели Кахонена

Основной целью карт самоорганизации, в рассматриваемой модели, является преобразование поступающих векторов сигналов, имеющих произвольную размерность, в одно- или двухмерную дискретную карту. При этом такое преобразование осуществляется адаптивно, в топологически упорядоченной форме. На рис. 2.5 показана схематическая диаграмма двумерной решетки нейронов, используемой в качестве дискретной карты.

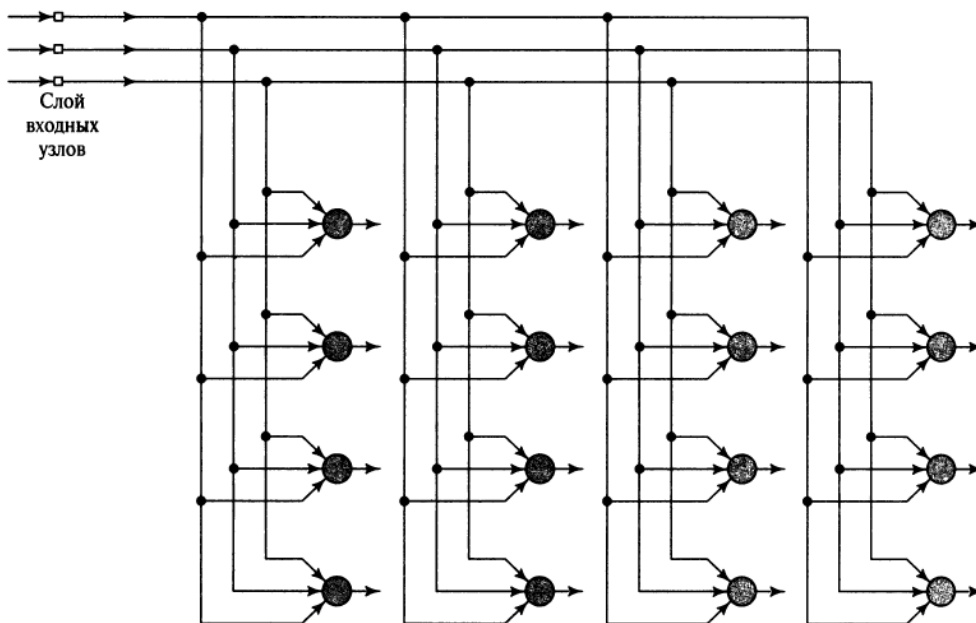


Рис. 2.5. Двумерная решетка нейронов, используемая в качестве дискретной карты

Все нейроны этой решетки связаны со всеми узлами входного слоя. Эта сеть имеет структуру прямого распространения с одним вычислительным слоем, состоящим из нейронов, упорядоченных в столбцы и строки.

Алгоритм, ответственный за формирование самоорганизующихся карт, начинается с инициализации синаптических весов сети. Обычно это происходит с помощью назначения синаптическим весам малых значений, сформированных генератором случайных чисел. При таком формировании карта признаков изначально не имеет какого-либо порядка признаков. После корректной инициализации сети для формирования карты самоорганизации запускаются три следующих основных процесса.

1. Конкуренция. Для каждого входного образа нейроны сети вычисляют относительные значения дискриминантной функции. Эта функция является основой конкуренции среди нейронов.

2. Кооперация. Победивший нейрон определяет пространственное положение топологической окрестности нейронов, обеспечивая тем самым базис для кооперации между этими нейронами.

3. Синаптическая адаптация. Последний механизм позволяет возбужденным нейронам увеличивать собственные значения дискриминантных функций по отношению к входным образам посредством соответствующих корректировок синаптических весов. Корректировки производятся таким образом, чтобы отклик нейрона-победителя на последующее применение аналогичных примеров усиливался.

Процессы конкуренции и кооперации осуществляются в соответствии с двумя из четырех принципов самоорганизации, описанных выше. Что же касается принципа самоусиления, то он реализуется в модифицированной форме - форме обучения Хебба для адаптивного процесса.

Опишем процессы присущие формированию самоорганизующихся карт.

1. Процесс конкуренции.

Выбирая нейрон с наибольшим скалярным произведением $w_j^T x$, мы в результате определяем местоположение, которое должно стать центром топологической окрестности возбужденного нейрона.

2. Процесс кооперации.

Нейрон-победитель находится в центре топологической окрестности сотрудничающих нейронов. Основным вопросом является механизм определения корректной топологической окрестности. Согласно принципам латерального взаимодействия, возбужденный нейрон всегда пытается возбудить наиболее близкие (в пространстве) к нему нейроны, что и приводит к определению топологической окрестности победившего нейрона j , которая уменьшается (теряет интенсивность) с увеличением расстояния, правило конкурентного обучения рассмотрено в работе Гроссберга [80] и работах [74], [75], [78].

3. Процесс адаптации

Для того чтобы сеть могла самоорганизоваться, вектор синаптических весов w_j ; нейрона j должен изменяться в соответствии с входным вектором x . В постулате обучения Хебба синаптический вес усиливался при одновременном возникновении предсинаптической и постсинаптической активности, что хорошо подходит к ассоциативному обучению и не подходит для типа обучения без учителя. Для преодоления данной проблемы введем дополнительное слагаемое забывания $g(y_j)w_j$, где w_j - вектор синаптических весов нейрона j ; $g(y_j)$ - некоторая положительная скалярная функция отклика y_j . Единственным требованием, налагаемым на функцию $g(y_j)$, является равенство нулю постоянного слагаемого разложения в ряд Тейлора функции $g(y_j)$, что, в свою очередь, влечет выполнение соотношения

$$g(y_j) = 0 \text{ при } y_j = 0. \quad (2.33)$$

Декомпозиция процесса адаптации синаптических весов сети, вычисляемая согласно (2.37), позволяет получить два этапа – этап самоорганизации и этап сходимости.

1. Этап самоорганизации и упорядочивания. На данном этапе происходит топологическое упорядочивание векторов весов. В рассматриваемом в работе алгоритме данный этап является затратным, его выполнение может потребовать до 10000 итераций, поэтому особое внимание необходимо уделить предварительной подготовке данных предвестников, выбору параметра скорости обучения и ширины функции окрестности.

Параметр скорости обучения, по результатам экспериментов, оказывается в пределах $0,01 \leq \eta(n) \leq 0,12$ (с учётом его убывания) для этого в выражении (2.38) константы η_0 и τ_2 равны 0,12 и 1000, соответственно.

2. Этап сходимости. Вторым этапом адаптивного процесса необходим для такой настройки карты признаков которая позволяет получить наиболее точное квантование входного пространства. Количество итераций, данного этапа, достаточное для этапа сходимости, примерно в 500 раз превышает количество нейронов сети. Таким образом, количество итераций этапа сходимости может достигать тысяч и десятков тысяч и является наиболее затратным по времени.

2.4. Описание алгоритма для выявления значимых метеорологических параметров

Сущность алгоритма SOM, предложенного Кохоненом, состоит в простом геометрическом вычислении свойств Хеббоподобного правила обучения и латеральных взаимодействий. Существенными характеристиками этого алгоритма являются следующие:

- непрерывное входное пространство образов активации, которые генерируются в соответствии с некоторым распределением вероятности;
- топология сети в форме решетки, состоящей из нейронов. Она определяет дискретное выходное пространство;
- зависящая от времени функция окрестности $h_{j,i}(n)$, которая определена в окрестности нейрона-победителя $i(x)$;
- параметр скорости обучения $\eta(n)$, для которого задается начальное значение η_0 и который постепенно убывает во времени n , но никогда не достигает нуля.

Для функции окрестности и параметра скорости обучения на этапе упорядочивания (т.е. приблизительно для первой тысячи итераций) можно использовать формулы (2.32) и (2.38) соответственно. Для хорошей статистической точности на этапе сходимости параметр $\eta(n)$ должен быть установлен в очень малое значение (0,01 или меньше). Что же касается функции окрестности, то в начале этапа сходимости она должна содержать только ближайших соседей нейрона-победителя (при этом может включать только его самого).

Не считая инициализации, алгоритм проходит три основных шага: подвыборка, поиск максимального соответствия и корректировка. Кратко весь алгоритм можно описать следующим образом.

1. Инициализация.
2. Подвыборка.
3. Поиск максимального подобия.
4. Коррекция.
5. Продолжение.

По завершении процесса сходимости алгоритма SOM вычисленная им карта признаков отображает важные статистические характеристики входного пространства.

Карта признаков Φ обладает следующими важными свойствами.

Свойство 1. Аппроксимация входного пространства. Карта признаков Φ , представленная множеством векторов синаптических весов $\{w_j\}$, в выходном пространстве A реализует хорошую аппроксимацию входного пространства X .

Свойство 2. Топологический порядок. Карта признаков Φ , полученная используемым алгоритмом, является топологически упорядоченной в том смысле, что пространственное положение нейронов в решетке соответствует конкретной местности с учётом соответствия признаков изменения МП признаку входных МД.

Свойство 3. Соответствие плотности. Карта признаков Φ отражает вариации в статистиках распределения входного сигнала. Области во входном пространстве X , из которого берутся векторы x с большей вероятностью, отображаются в гораздо большие области выходного пространства A и, таким образом, с большим разрешением, чем области в исходном пространстве X , из которых берутся векторы x с меньшей вероятностью.

Свойство 4. Выбор признаков. Для данных из входного пространства с нелинейным распределением самоорганизующаяся карта для аппроксимации исследуемого распределения способна извлечь набор наилучших признаков.

Это свойство является естественным обобщением свойств 1-3. Данное свойство наиболее ярко выражается при использовании АК на предварительном этапе подготовки данных. Однако это свойство имеет одно важное отличие, которое показано на рис. 2.8.

2.5. Компьютерное моделирование и анализ нейронной сети

В качестве исследуемой использовалась сеть, состоящая из 100 нейронов, упорядоченных в форме двумерной решётки размером 10x10. Данная сеть обучается на двумерных входных векторах x , элементы x_1 и x_2 которых равномерно распределены в областях $\{(-1 < x_1 < +1); (-1 < x_2 < +1)\}$. Для инициализации сети синаптические веса выбираются случайным образом.

На рис. 2.9 показаны три этапа обучения сети представлению входного распределения. На рис. 2.9, *а* показано равномерное распределение данных, использованных для обучения карты признаков. На рис. 2.9, *б* показаны исходные значения случайно выбранных синаптических весов.

На рис. 2.9, *в*, *г* показаны значения векторов синаптических весов после завершения этапов упорядочивания и сходимости. Синаптические веса на этих рисунках показаны точками. Линии, показанные на рис. 2.9, соединяют соседние нейроны (вдоль строк и столбцов) сети.

Результаты, демонстрируют этапы упорядочивания и сходимости, которые характеризуют процесс обучения по алгоритму SOM. Во время этапа упорядочивания карта напоминает форму сетки (см. рис. 2.9, *в*). К концу этого этапа нейроны отображаются в правильном порядке. Во время этапа сходимости сетка пытается «накрыть» все входное пространство. В конце второго этапа (см. рис. 2.9, *г*) статистическое распределение нейронов на карте приближается к распределению входных векторов (за исключением некоторых граничных эффектов).

Сравнивая конечное состояние карты признаков (см. рис. 2.9, *г*) с равномерным распределением входных сигналов (см. рис. 2.9, *а*), мы видим, что точная настройка карты, выполненная на этапе сходимости, выявила отдельные локальные неравномерности, присутствующие во входном распределении.

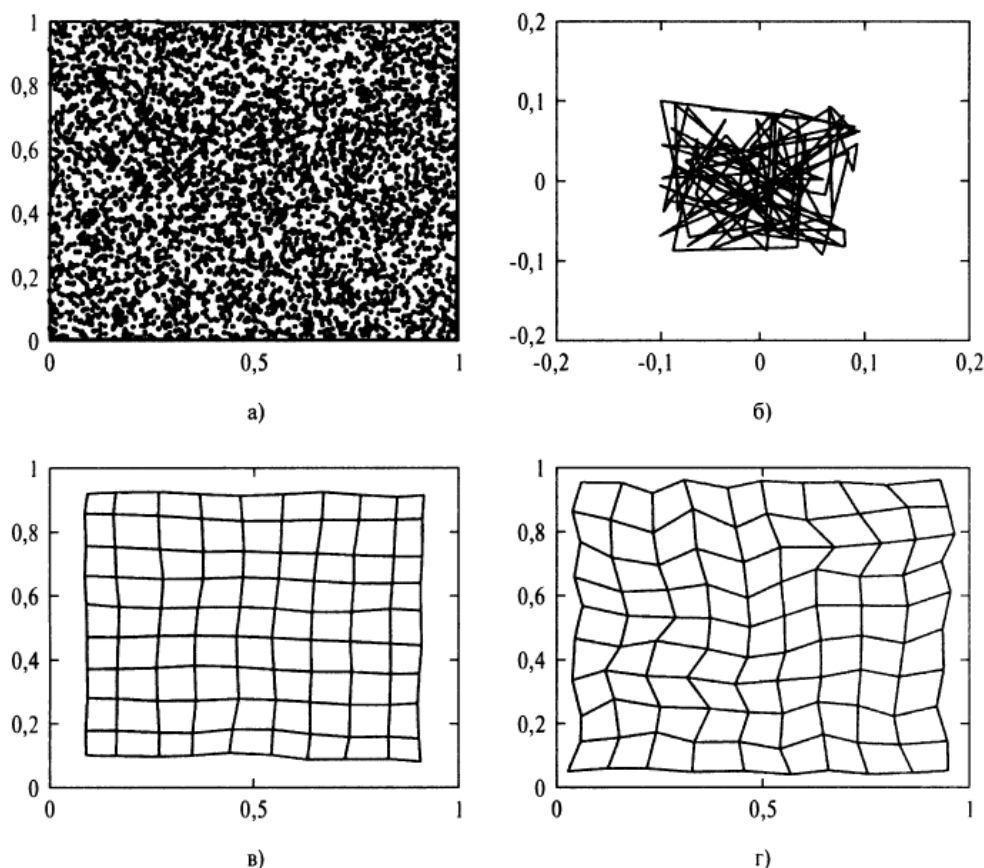


Рис. 2.9. Распределение входных данных (а); начальное состояние двумерной решетки (б); состояние решетки в конце этапа упорядочивания (в); состояние решетки в конце этапа сходимости (г)

Свойство топологического упорядочивания алгоритма SOM хорошо продемонстрировано на рис. 2.9, г. В частности, видно, что алгоритм (после сходимости) извлек топологию распределения входного сигнала. В результатах компьютерного моделирования (см. рис. 2.9) как входное пространство X , так и выходное пространство A было двумерным.

Рассмотрим случай, когда размерность входного пространства X больше размерности выходного пространства A . Несмотря на это расхождение, карта признаков Φ часто способна сформировать топологическое представление входного распределения. На рис. 2.10 показаны три этапа эволюции карты признаков, инициализированной на рис. 2.10, б и обучаемой на входных данных, взятых из равномерного распределения внутри квадрата (см. рис. 2.9, а). Вычисления осуществлялись на одномерной решетке, состоящей из 100 нейронов.

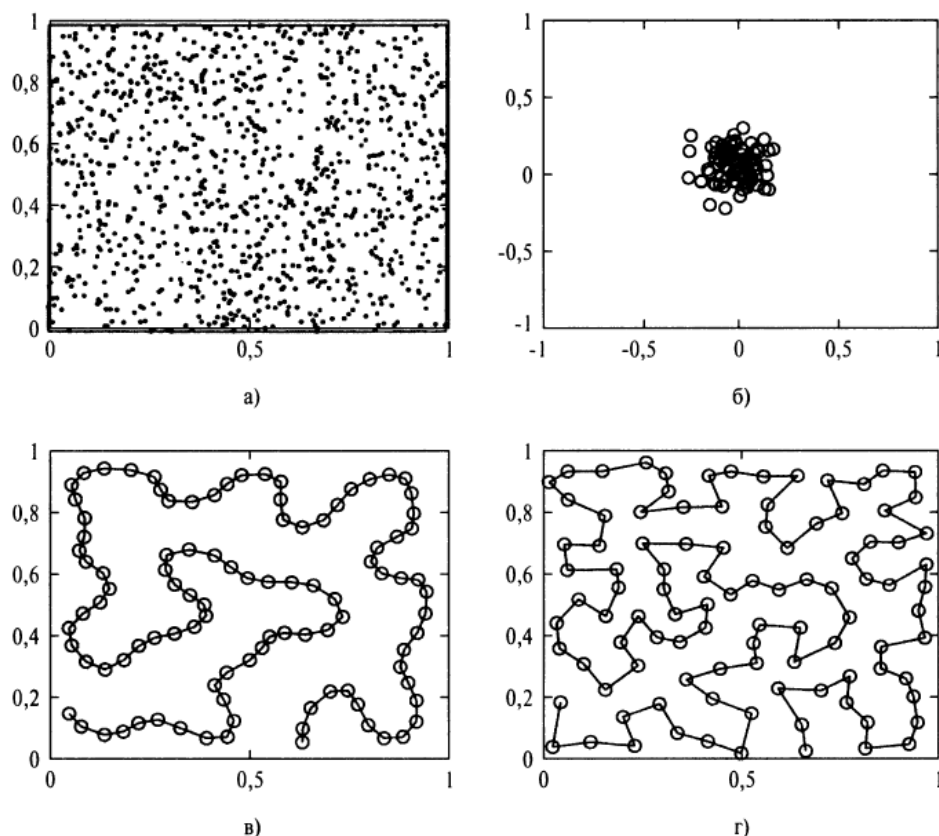


Рис. 2.10. Распределение входных данных (а); начальное состояние двумерной решетки (б); состояние решетки в конце этапа упорядочивания (в); состояние решетки в конце этапа сходимости (г)

На рис. 2.10, в, г показаны карты признаков после завершения этапов упорядочивания и сходимости соответственно. На этих рисунках видно, что карта признаков, вычисленная алгоритмом, не может корректно представить равномерное заполнение квадрата достаточно плотно и дать достаточно хорошую аппроксимацию двумерного входного пространства X . Операция, проведенная в этом эксперименте и представленная картой признаков на рис. 2.10 (где входное пространство X представлено его проекцией на пространство более низкой размерности A), можно охарактеризовать как снижение размерности.

На рис. 2.11 представлены детали динамики функции окрестности $h_{j,i}(n)$ и параметра скорости обучения $\eta(n)$ с течением времени (т.е. с количеством эпох) при ретроспективном моделировании отображения сигнала (метеорологического параметра) на одномерную решетку.

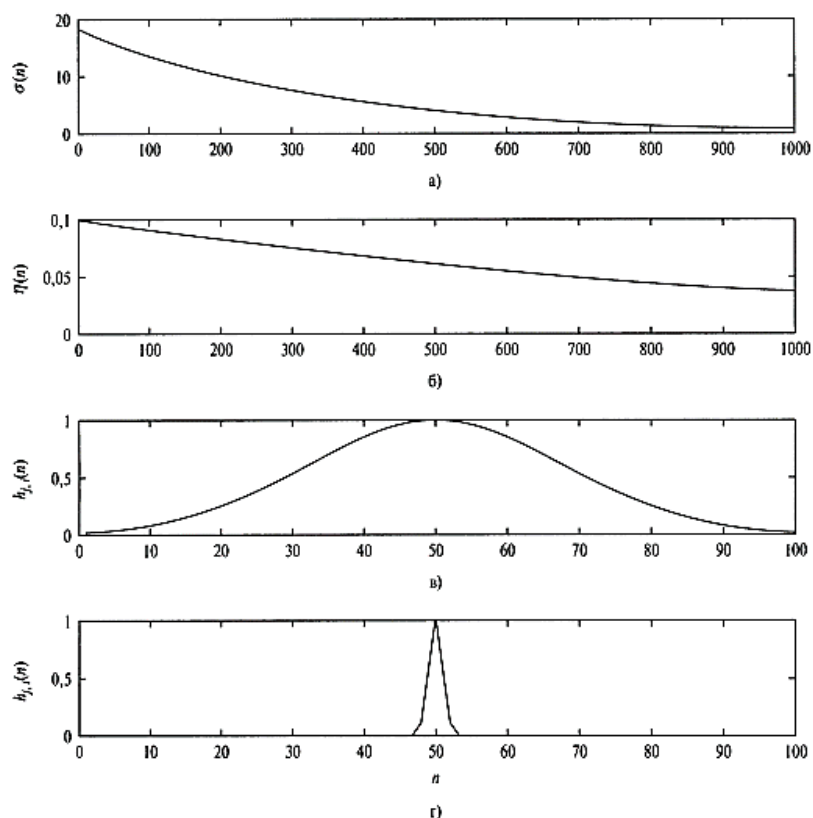


Рис. 2.11. Экспоненциальное убывание параметра $\sigma(n)$ функции окрестности (а); экспоненциальное убывание параметра $\eta(n)$ (б); изначальная форма гауссовой функции окрестности (в); форма функции окрестности по завершении фазы упорядочивания (т.е. в начале этапа сходимости) (г)

Параметр функции окрестности $\sigma(n)$, показанный на рис. 2.11, а, начинает работу из исходного состояния $\sigma_0 = 18$ и убывает практически до единицы за 1000 итераций этапа упорядочивания.

В течение того же этапа параметр скорости обучения $\eta(n)$ начинает работу со значения 0,1 и уменьшается до значения 0,037. На рис. 2.11, в показано изначальное гауссово распределение нейронов вокруг победившего нейрона, расположенного в центре одномерной решетки. На рис. 2.10, г показана форма функции окрестности в конце этапа упорядочивания. Во время этапа сходимости (5000 итераций) параметр скорости обучения равномерно уменьшался от 0,037 до 0,001. В течение того же этапа функция окрестности сократилась практически до нуля.

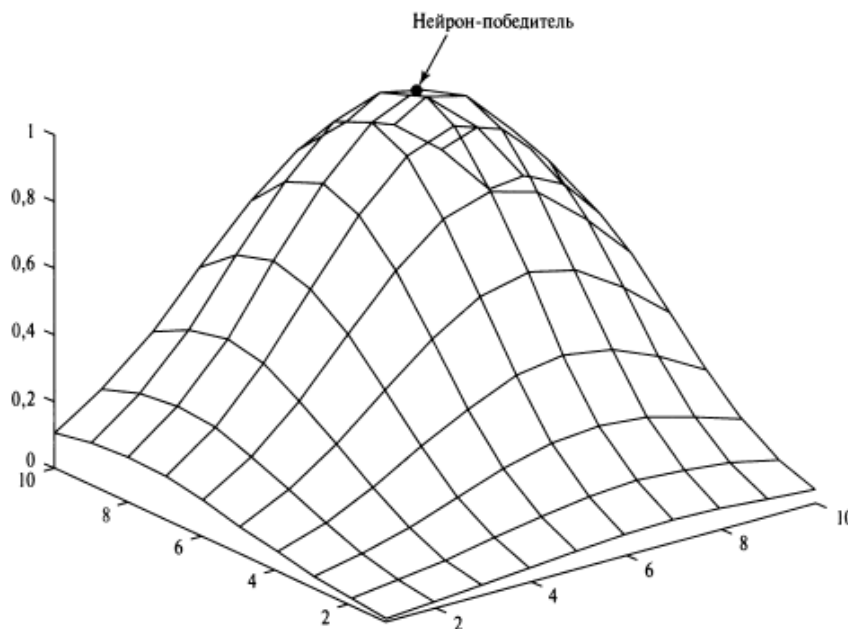


Рис. 2.12. Начальное условие для двумерной гауссовой функции окрестности с центром в нейроне-победителе, расположенном в точке (7; 8) двумерной решетки размером 10x10 нейронов

Спецификации этапов упорядочивания и сходимости при компьютерном моделировании ретроспективного сигнала МД для случая двумерной решетки (см. рис. 2.9) аналогичны использованному для одномерной, за исключением того факта, что функция окрестности теперь тоже стала двумерной. Параметр $\sigma(n)$ начал работу со значения $\sigma_0 = 3$, за 1000 итераций уменьшившись до 0,75. На рис. 2.12 показано начальное состояние гауссовой функции окрестности $h_{j,i}(n)$ для $\sigma_0 = 3$ и победившего нейрона, центрированного в точке (7; 8) двумерной решетки размером 10 x 10 нейронов.

2.6. Алгоритм квантования вектора обучения

Векторное квантование представляет собой прием использования структуры входных векторов для сжатия данных [88]. В частности, входное пространство подразделяется на множество обособляемых (по признакам) областей, для каждой из которых определяется вектор воспроизведения. Когда устройству квантования представляется новый входной вектор, в первую очередь определяется область, к которой принадлежит данный вектор, а после этого создается представление через вектор воспроизведения данной области. Таким образом, используя некодированную версию восстанавливаемого вектора для хранения или передачи исходного вектора, можно добиться значительной экономии пространства хранения или мощности канала передачи данных за счет

привнесения некоторых искажений. Множество возможных векторов воспроизведения называется книгой кодировки устройства квантования, а его отдельные члены - кодовыми словами.

Устройство векторного квантования, обладающее минимальным искажением при кодировании, называется квантователем Вороного, или квантователем на основе ближайшего соседа, а ячейками Вороного называют множество точек во входном пространстве, которые соответствуют подразделению этого пространства в соответствии с правилом ближайшего соседа, основанным на Евклидовой метрике [88]. На рис. 2.13 показан пример входного пространства, разделенного на четыре ячейки Вороного, с соответствующими векторами воспроизведения (или векторами Вороного). Каждая из этих ячеек содержит те точки входного пространства, которые расположены ближе всего к вектору Вороного.

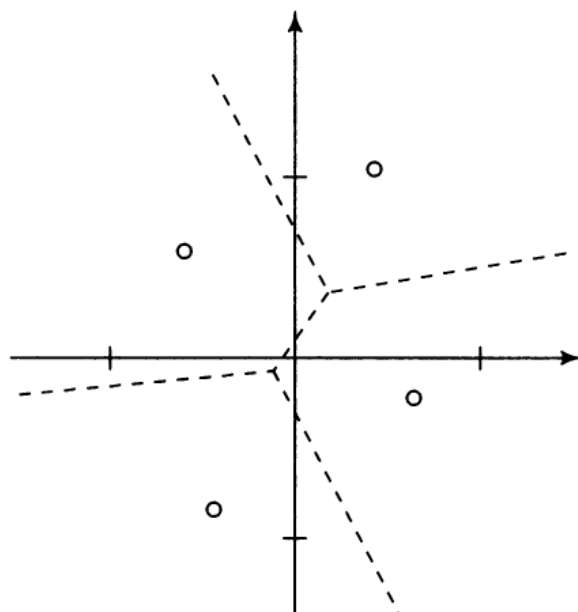


Рис. 2.13. Диаграмма Вороного, содержащая 4 ячейки

Алгоритм SOM обеспечивает приближенный метод вычисления векторов Вороного без учителя. При этом аппроксимация определяется векторами синаптических весов нейронов на карте признаков. Это практически повторяет описание свойства 1 алгоритма SOM, о котором речь шла в разделе 2.1. Расчет карты признаков, таким образом, можно рассматривать как первый из двух этапов адаптивного решения задачи классификации (рис. 2.14). На втором этапе

в качестве механизма точной подстройки карты признаков проводится квантование вектора обучения.

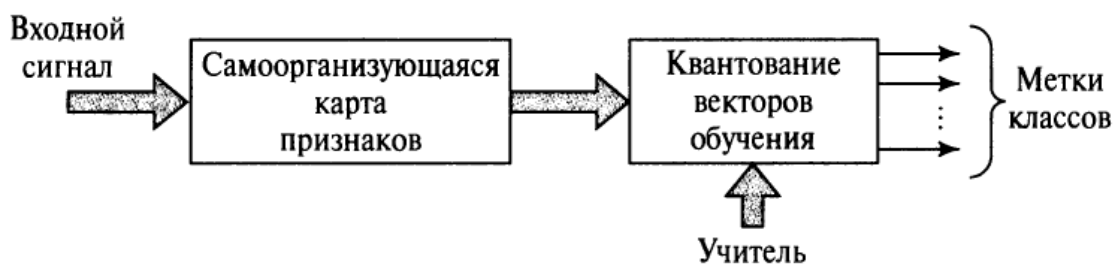


Рис. 2.14. Блочная диаграмма адаптивной классификации множеств, использующей самоорганизующуюся карту признаков и квантование вектора обучения

Квантование вектора обучения[†] - это прием обучения с учителем, который использует информацию о классе для небольшого смещения вектора Вороного и, таким образом, для улучшения качества областей решений классификатора.

Входной вектор x случайно выбирается из входного пространства. Если метки класса входного вектора x и вектора Вороного w согласуются, последний смещается в направлении первого. Если же метки классов этих векторов не согласуются, вектор Вороного w смещается в сторону, противоположную входному вектору x .

Пусть $\{w_j\}_{j=1}^l$ - множество векторов Вороного, а $\{x_i\}_{i=1}^N$ множество входных векторов (наблюдений). Предполагается, что количество входных векторов намного превосходит количество векторов Вороного, что, как правило, происходит на практике. Алгоритм квантования векторов обучения (LVQ1) можно описать следующим образом.

1. Предположим, что вектор Вороного w_c является самым близким к входному вектору x_i . Обозначим символом C_{w_c} класс, ассоциируемый с вектором Вороного w_c , а символом C_{x_i} - метку класса входного вектора x_i . Вектор Вороного w_c корректируется следующим образом.

1. Если $C_{w_c} = C_{x_i}$ то

[†] Идея квантования вектора обучения принадлежит Кохонену [578]. Три версии этого алгоритма были описаны в его работах [568], [574]. Версия, предлагаемая в работе, является первой из них и в работах Кохонена обозначается как LVQ1.

Алгоритм квантования вектора обучения является алгоритмом стохастической аппроксимации. В работе [89] исследовались свойства сходимости этого алгоритма. При этом использовался подход ODE (обычных дифференциальных уравнений), который описывался в главе 1.

$$w_c(n + 1) = w_c(n) + \alpha_n[x_i - w_c(n)], \text{ где } 0 < \alpha_n < 1.$$

2. Если $C_{w_c} \neq C_{x_i}$, то

$$w_c(n + 1) = w_c(n) - \alpha_n[x_i - w_c(n)].$$

2. Остальные векторы Вороного не изменяются.

Используемая в алгоритме константа обучения α_n задаётся монотонно убывающей с увеличением числа итераций.

2.7. Адаптивная классификация множеств

В задаче классификации предварительно подготовленных множеств первым и наиболее важным шагом является извлечение признаков, которое выполняется без учителя. Целью этого шага является выбор обоснованно малого количества признаков, в которых сконцентрирована самая существенная информация о входных (классифицируемых) МД. Для извлечения признаков лучше всего подходит самоорганизующаяся карта, ввиду ее свойства 4 (см. п. 2.1), особенно учитывая, что множества входных данных формируются нелинейным процессом.

Вторым шагом в задаче классификации множеств является сама классификация, в которой признакам, выделенным на первом шаге, назначаются разные классы. Несмотря на то что и самоорганизующаяся карта может самостоятельно осуществлять классификацию, все же для повышения производительности алгоритм будет дополнен схемой обучения с учителем. Комбинация самоорганизующейся карты со схемой обучения с учителем формирует базис адаптивной классификации множеств, которая является гибридной по своей сущности.

Гибридный подход к классификации достигается использованием квантованием вектора обучения, как было описано выше. Исходя из сказанного, получим двухступенчатый адаптивный классификатор (см. рис. 2.15).

На рис. 2.15, а показана двумерная карта признаков, состоящая из решетки размером 5x5 нейронов после завершения работы алгоритма SOM.

Эта карта была маркирована, и каждый из нейронов был поставлен в соответствие одному из предвестников, в зависимости от того, как он

среагировал на тестовые данные, взятые из входного распределения. На рис. 2.15, б показана граница решений, сформированная самой картой признаков.

На рис. 2.15, в показана модифицированная карта признаков, настроенная в процессе обучения с учителем на основе алгоритма DBSCAN. На рис. 2.15, г показана граница решений, полученная в результате совместного действия алгоритмов SOM и DBSCAN. Сравнивая последние два рисунка с их двойниками, 2.15, а и 2.15, б, можно увидеть качественный эффект применения алгоритма DBSCAN.

В табл. 2.1 представлены эффективности классификации для карты признаков, действующей самостоятельно, и для совмещения алгоритмов SOM и DBSCAN. Представленные здесь результаты получены из 10 независимых пробных эксперимента, в каждой из которых использовалось по 30000 примеров тестовых данных.

График распределения данных, используемых при проведении моделирования, показан на рис. 2.16.

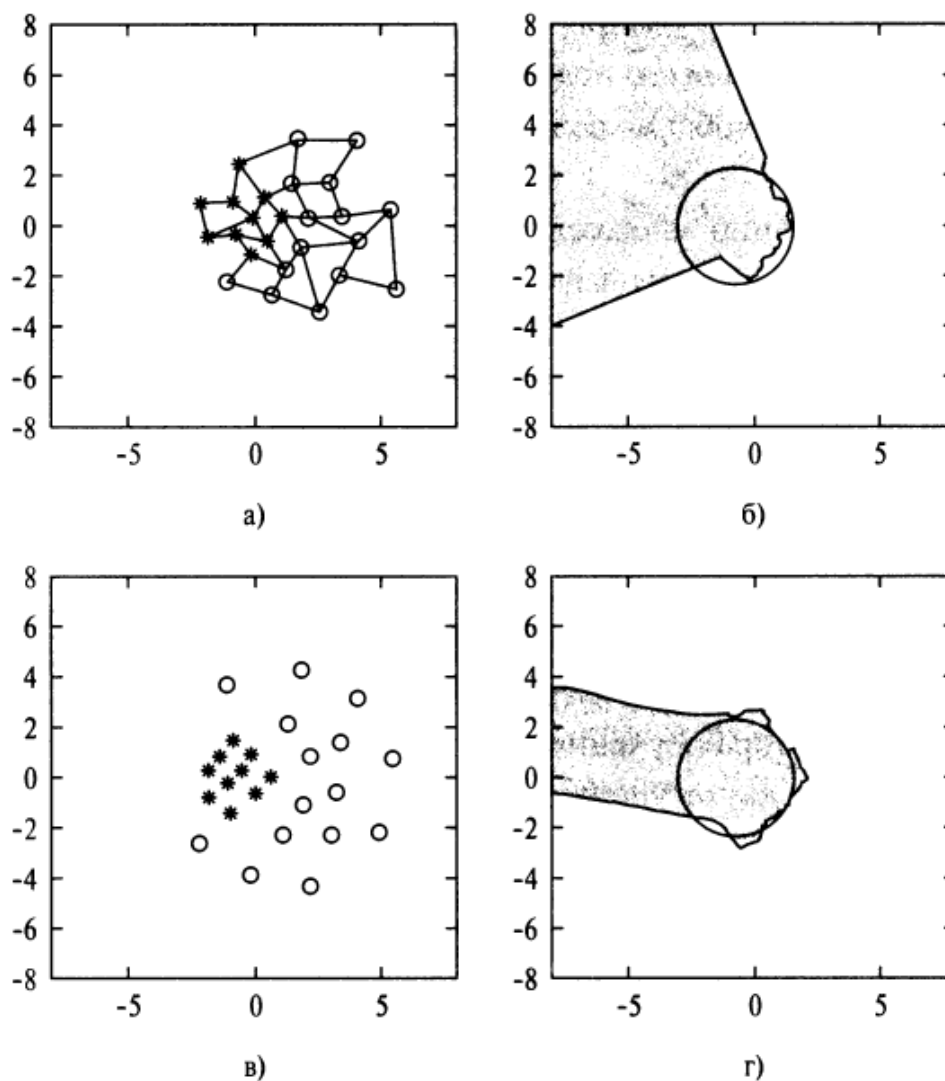


Рис. 2.15. Самоорганизующаяся карта после расстановки меток (а). Граница решений (б), построенная картой (а). Маркированная карта после квантования векторов обучения (в). Граница решений (г), построенная на карте (в)

В каждом эксперименте наблюдалось повышение качества классификации при применении алгоритма DBSCAN. Средний показатель производительности для карты признаков, действующей самостоятельно, составил 79,61%, а для комбинации двух алгоритмов - 80,52%, т.е. среднее улучшение качества классификации составило 0,91%.

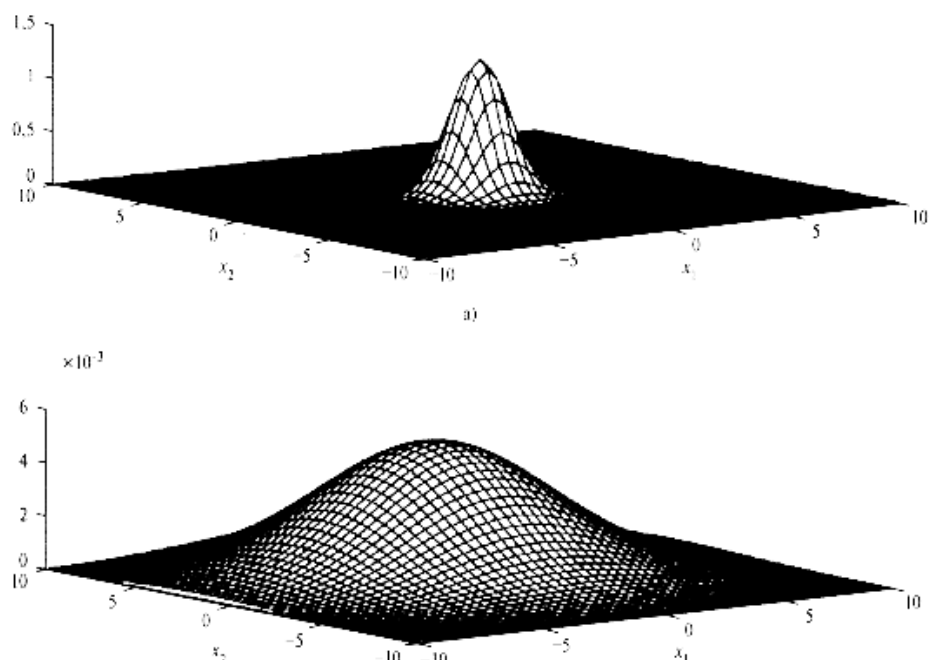


Рис. 2.16. Функции плотности вероятности для эксперимента

Для сравнения - производительность оптимального классификатора Байеса [56] для этого эксперимента составила 81,51 %.

Табл. 2.1. Эффективность классификации (в процентах)

Попытка	Самостоятельная классификация картой признаков	Классификация комбинацией карты признаков и DBSCAN
1	79,05	80,18
2	79,79	80,56
3	79,41	81,17
4	79,38	79,84
5	80,30	80,43
6	79,55	80,36
7	79,79	80,86
8	78,48	80,21
9	80,00	80,51
10	80,32	81,06
Среднее	79,61%	80,52%

2.8. Иерархическое квантование векторов

Необходимо отметить что наилучшего сжатия данных всегда можно добиться с помощью кодирования не векторов, а скаляров, даже если источник данных не имеет памяти (т.е. является последовательностью независимых случайных переменных), или если система сжатия данных имеет память (т.е. действия кодировщика зависят от своих предыдущих входов или выходов) [68].

Однако алгоритмы обычного векторного квантования требуют довольно большого объема вычислений, что ограничивает их практическое

использование. Большую часть времени в алгоритме векторного квантования занимает операция кодирования. При кодировании входной вектор должен сравниваться с каждым из векторов кодирования в кодовой книге, для того чтобы найти тот, который обеспечивает минимальное искажение. Например, если кодовая книга содержит N векторов кодирования, время, затраченное на кодирование, будет иметь порядок N . Исходя из этого, с увеличением N оно будет возрастать. В [81] описано многоступенчатое иерархическое кодирование, которое будет применено при сжатии исходных МД, что значительно увеличивает скорость кодирования. Используемый алгоритм иерархического кодирования является многоступенчатым иерархическим векторным квантователем, использующим принцип разложения операции векторного квантования на множество подопераций, каждая из которых требует малого (по сравнению с исходным) объема вычислений. Используя механизм подготовки данных (квантование), описанный в разделе 1 и модифицированный алгоритм SOM для обучения каждой из подопераций, можно добиться малой потери точности (до доли децибела) при одновременном повышении скорости вычислений.

Рассмотрим работу двухступенчатого векторного квантователя, VQ_1 и VQ_2 , использующегося при оценке МД. Первый передает свой выход на второй. Выход второго квантователя является окончательной кодированной версией исходного входного сигнала, примененного к VQ_1 . При осуществлении квантования необходимо добиться того, чтобы VQ_2 не обеднял (учитывал все статистически значимые данные) исходную информацию. Поскольку рассматривается квантователь VQ_1 исключительный эффект от VQ_2 заключается в искажении выхода VQ_1 . Таким образом, получается, что для VQ_1 больше подходит алгоритм SOM, который учитывает искажение сигнала, налагаемое квантователем VQ_2 [81]. Для того чтобы использовать обобщенный алгоритм Ллойда для обучения квантователя VQ_2 , нужно предположить, что выход VQ_2 не будет искажен до проведения восстановления. В этом случае не потребуется вводить зашумленную модель (для выхода VQ_2) и ассоциированную с ней функцию окрестности конечной ширины.

Обобщение этого аргумента на многоступенчатый векторный квантователь позволяет использовать следующее правило - каждый шаг должен учитывать искажения, внесенные всеми предыдущими шагами, и моделировать их как шум. Поэтому для обучения всех ступеней квантователя (за исключением последней, для которой применяется обобщенный алгоритм Ллойда) используется алгоритм SOM.

Иерархическое векторное квантование является частным случаем многоступенчатого векторного квантования [81]. В качестве иллюстрации рассмотрим квантование входного вектора размерности 4×1 : $x = [x_1, x_2, x_3, x_4]^T$.

На рис. 2.17, *а* показан одноступенчатый векторный квантователь для x . В качестве альтернативы используется двухступенчатый иерархический квантователь, представленный на рис. 2.17, *б*. Кардинальное отличие между двумя этими схемами состоит в том, что размерность входа квантователя на рис. 2.17, *а* равна четырем, в то время как квантователя на рис. 2.17, *б* - двум. Соответственно квантователь, представленный на рис. 2.17, *б*, требует для своей работы существенно меньшей справочной таблицы и является более простым для реализации, чем квантователь, показанный на рис. 2.17, *а*. В этом и лежит главное преимущество иерархического квантователя перед обычным [61].

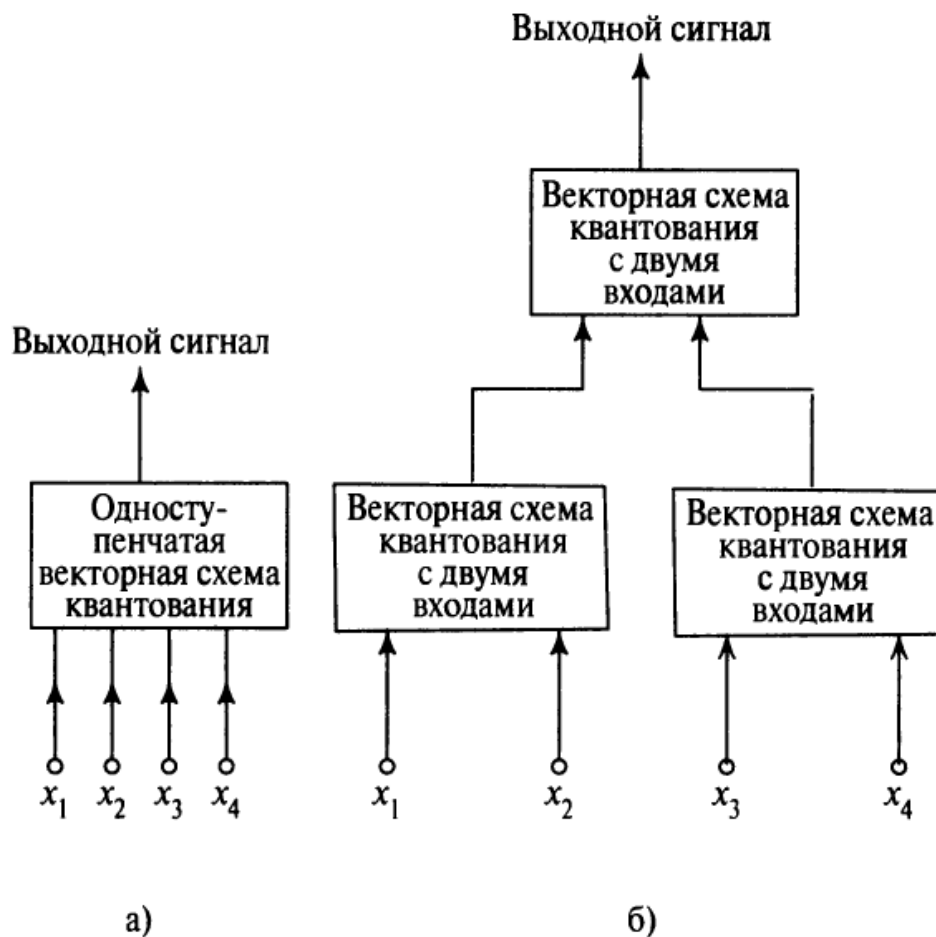


Рис. 2.17. Одноступенчатый векторный квантователь с четырехмерным входным сигналом (а); двухступенчатый иерархический векторный квантователь, использующий двухвходовые векторные квантователи (б)

На рис. 2.18 воспроизведены результаты, полученные в этой работе, для случая коррелированного гауссова шума, генерируемого с помощью используемого в работе метода: $x(n+1) = \rho x(n) + v(n)$,

где ρ - коэффициент авторегрессии; $v(n)$, - независимая и равномерно распределенная гауссова случайная переменная с нулевым средним и единичной дисперсией.

Исходя из этого, можно показать, что $x(n)$ характеризуется следующим образом: $E[x(n)] = 0, E[x^2(n)] = \frac{1}{1-\rho^2}, \frac{E[x(n+1)x(n)]}{E[x^2(n)]} = \rho$.

Таким образом, ρ можно рассматривать как коэффициент корреляции временного ряда $\{x(n)\}$. Чтобы инициировать генерацию этого временного ряда, для $x(0)$ использована гауссова случайная переменная с нулевым средним и

дисперсией $\frac{1}{1-\rho^2}$, а в качестве коэффициента корреляции выбрана величина $\rho = 0,85$.

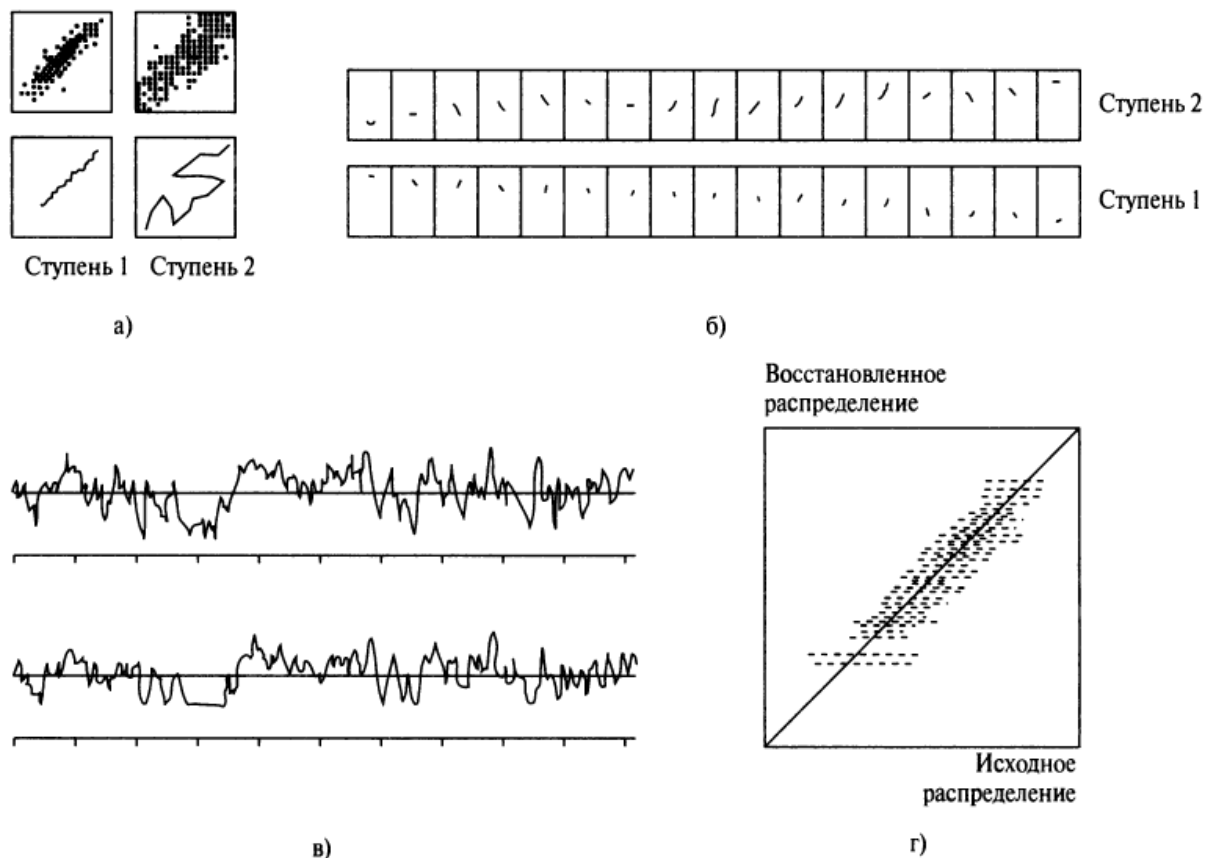


Рис. 2.18. Результаты двухступенчатого кодирования/декодирования для коррелированного гауссова шума. Коэффициент корреляции $\rho = 0,85$

Для векторного квантования использовался алгоритм АК с четырехмерным входным пространством, подобный показанному на рис. 2.17, б. Для ряда авторегрессии $\{x(n)\}$ потребовалась симметрия преобразования, подразумевающая использование только двух отдельных справочных таблиц. Размер каждой из этих таблиц экспоненциально зависел от количества битов входного сигнала и линейно от количества битов выходного сигнала. Во время обучения для правильного вычисления коррекций потребовалось большое количество битов для представления чисел, поэтому во время обучения справочные таблицы не использовались. После завершения обучения количество битов могло быть уменьшено до обычного уровня, после чего потребовались записи справочных таблиц. Для кодера, показанного на рис. 2.17, б, входные образы аппроксимировались с использованием четырех битов на образ. Для всех ступеней кодера использовалось $N=17$ векторов кодирования, так что количество

выходных битов в каждой из справочных таблиц приблизительно равнялось четырем. Таким образом, размер адресного пространства для таблиц обеих ступеней составил $256 (= 2^{4+4})$. Это говорит о том, что общие требования к памяти для представления таблиц были небольшими.

На рис. 2.18 показаны результаты кодирования/декодирования элементов $x(n)$. В нижней части рис. 2.18, *а* векторы кодирования для каждой из двух ступеней показаны как кривые, помещенные в двумерное входное пространство. В верхней части рис. 2.18, *а* показаны оценки соответствующих матриц соответствия размером 16×16 . На рис. 2.18, *б* в виде фрагментов временного ряда представлены следующие элементы:

- вектор кодирования, вычисленный первой ступенью кодера;
- вектор восстановления, вычисленный второй ступенью, который минимизирует среднеквадратическое искажение при одновременном сохранении всех остальных переменных.

На рис. 2.18, *в* представлено 512 примеров из исходного временного ряда (верхняя кривая) совместно с реконструированными примерами (нижняя кривая). Эти данные взяты из выхода последней ступени кодировщика. Горизонтальная шкала на рис. 2.18, *в* равна половине шкалы на рис. 2.18, *б*. На рис. 2.18, *г* представлена матрица соответствия, составленная из пар примеров - исходных и соответствующих им восстановленных. Ширина полосы на рис. 2.18, *г* показывает степень искажения, произведенного иерархическим векторным квантованием.

Исследуя графики на рис. 2.18, *в*, можно увидеть, что восстановленная информация является достаточно хорошим представлением исходного временного ряда, за исключением того, что отдельные положительные и отрицательные пики срезаны. Нормированное среднеквадратическое отклонение составило 0,15, что практически не хуже (меньше на 0,05 дБ), чем 8,8 дБ, полученные с использованием одноступенчатого блочного кодера с четырьмя входами, использующего по одному биту на пример.

2.9. Отображение результатов анализа

Существуют два фундаментально отличных метода визуализации самоорганизующихся карт признаков. Первый из этих методов состоит в построении гибкой сети, в которой векторы синаптических весов являются указателями, направленными от соответствующих нейронов во входное пространство. Этот метод визуализации особенно полезен для правильного отображения свойства топологического упорядочивания алгоритма SOM. Это было продемонстрировано на примере компьютерного моделирования, приведенного в разделе 2.6.

Во втором методе визуализации нейронам в двумерной решетке (представляющим выходной слой сети) назначаются метки классов, в зависимости от того, как каждый из примеров (не встречавшихся ранее) возбудил конкретный нейрон в самоорганизующейся сети. В результате этой второй ступени моделирования нейроны в двумерной решетке разбиваются на некоторое количество когерентных областей. Под когерентностью понимается то, что каждая из групп нейронов представляет обособленное множество непрерывных символов или меток [69]. Это подразумевает, что при восстановлении хорошо упорядоченной карты использовались правильные условия.

Для примера рассмотрим множество данных, представленных в табл. 2.2. В этой таблице приведены характеристики отдельных МП по выбранному региону. На основании данных каждого из столбцов таблицы можно составить метеорологический прогноз для выбранного региона по представленным МД с учётом изменений ретроспективного периода. МД, полученные из открытых источников, обозначились следующим образом: МП1 - средняя скорость горизонтального ветра в м/с; МП2 – среднее направление горного ветра в °С; МП3 – минимальная скорость мгновенного горного ветра в м/с; МП4 - максимальная, скорость мгновенного горного ветра в м/с; МП5 – средняя скорость вертикального ветра в м/с; МП6 – атмосферное давление в мм. рт. ст.; МП7 – относительная влажность воздуха в %; МП8 – температура точки росы в °С; МП9 - упругость (давление) водяного пара в гПа; МП10 - абсолютная

влажность в г/м³; МП11 - плотность воздуха в г/м³; МП12 - скорость звука в воздухе в м/с.

Табл. 2.2. Характеристики отдельных МП по региону

Ретро-й анализ МП	Значения отличий МП, в %	М П1	М П2	М П3	М П4	М П5	М П6	М П7	М П8	М П9	МП 10	МП 11	МП 12
Понижались	МП 2%	0	0	1	0	1	0	1	0	1	0	0	0
	МП 10%	0	0	1	1	1	0	1	0	1	0	0	0
	МП 20%	0	0	1	1	1	0	1	0	1	0	0	0
	МП 50%	0	1	0	0	0	1	0	0	0	0	1	1
	МП 80%	0	1	0	1	1	0	1	1	0	0	1	0
Повышались	МП 2%	0	0	1	1	1	0	1	0	1	0	0	0
	МП 10%	0	1	0	0	0	1	0	0	0	0	1	1
	МП 20%	0	0	1	1	1	0	1	0	1	0	0	0
	МП 50%	0	0	1	1	1	0	1	0	1	0	0	0
	МП 80%	0	0	1	1	1	0	1	0	1	0	0	0

При этом значение 1 подразумевает наличие, а 0 - отсутствие (понижения, повышения) одного из 12 метеорологических параметров, перечисленных в столбце слева. Отдельные МП (например, «МП7» и «МП8») коррелируют, в то время как остальные - нет. Для каждого из анализируемых регионов можно составить характеристический код x_a , состоящий из двенадцати элементов. Регион определяется символьным кодом x_s , по которому никак нельзя собрать информацию о сходствах и различиях между отдельными МП. Например, в данном примере x_s представляет собой вектор, k -й элемент (соответствующий номеру МП в списке) которого имеет некоторое значение a - при этом все остальные элементы равны нулю. Параметр a определяет относительное влияние характеристического и символьного кодов друг на друга. Для того чтобы гарантировать, что данный характеристический код является доминантным, a выбирается равным 0,2. Входной вектор для каждого из МП состоит из 29 элементов и представляет собой объединение характеристического x_a и символьного x_s кодов:

$$x = \begin{bmatrix} x_s \\ x_a \end{bmatrix} = \begin{bmatrix} x_s \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ x_a \end{bmatrix}.$$

Наконец, каждый из векторов данных нормируется к единичной длине. Примеры из таким образом созданного множества данных подаются на вход

двумерной решетки нейронов размером 10x10, при этом синаптические веса нейронов настраиваются в соответствии с алгоритмом SOM (см. рис. 2.1). Обучение продолжалось в течение 2000 итераций, после чего карта признаков должна была достичь стабильного состояния. После этого в самоорганизующейся сети подавались на вход векторы $x = [x_s, 0]^T$, содержащие только символьные коды МП, и определялся нейрон с самым сильным откликом. Эта процедура повторялась для всех двенадцати МП.

В ней маркированные нейроны дали самый сильный отклик на соответствующие примеры. Точками на карте обозначены нейроны с более слабыми откликами.

В процессе выполнения описанного алгоритма была получена карта, показанная на рис. 2.19.

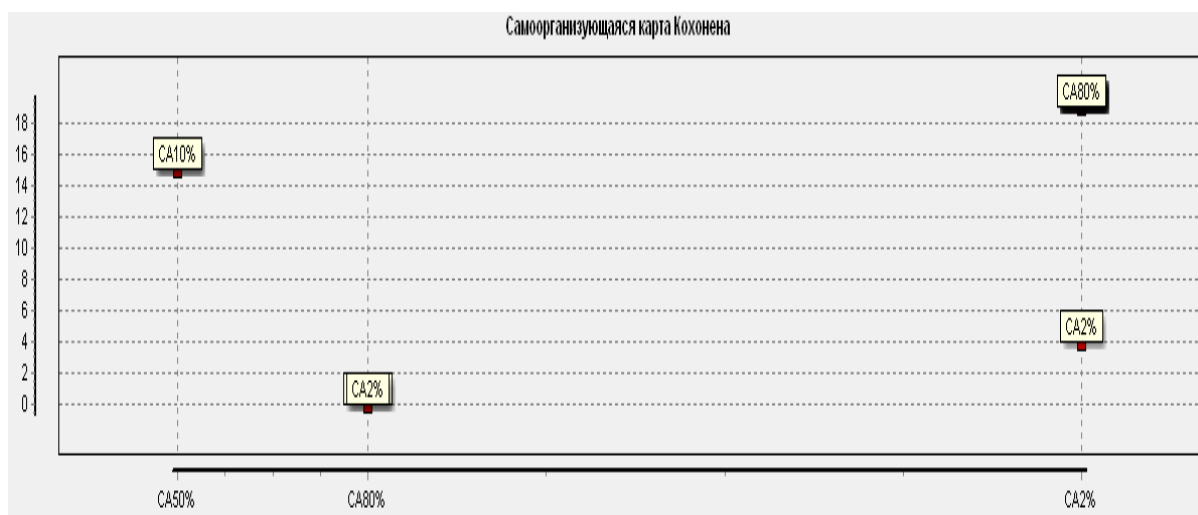


Рис. 2.19. Карта признаков, содержащая маркированные нейроны, имеющие самый сильный отклик на определенный входной пример

На рис. 2.20 показан результат моделирования для той же самоорганизующейся сети. Однако на этот раз каждый из нейронов сети был маркирован названием того МП, для которого его отклик был самым сильным. На этом рисунке видно, что карта признаков четко отслеживает «родственные взаимосвязи» между 12 различными признаками. На нем можно выделить три обособленных кластера, один из которых представляет регионы с повышенным значением, второй – со средней, а третий – с низким значением выбранного МП. Необходимо учесть, что значения коэффициентов преднамеренно устанавливались в бинарное значение, для определения взаимосвязи

кластеризации. Реальные значения коэффициентов параметров, лежат в более широких диапазонах и распределены так, что их анализ позволяет однозначно определить вероятности изменения в большую или меньшую сторону.

Карта признаков, показанная на рис. 2.20, называется контекстной или семантической [102], [89]. Полученные в работе контекстные карты, получившиеся в результате использования алгоритма SOM, могут применяться в решении задач анализа каталогов МД или в исследовании МП и извлечении из них скрытых закономерностей [75].



Рис. 2.20. Семантическая карта, полученная при использовании моделирования.

Предложенный в работе алгоритм анализа МП на основе самоорганизующиеся карты представляет собой нейронную сеть, построенную на одно- или двухмерной решетке нейронов для извлечения важных признаков, содержащихся во входном пространстве. Таким образом, можно получить структурное представление входных данных, используя в качестве прототипов векторы весов нейронов. Алгоритм SOM имеет нейробиологическую подоплеку. Он объединил в себе все основные механизмы, присущие самоорганизации: конкуренцию, кооперацию и самоусиление (см. главу 1). Таким образом, его можно рассматривать как универсальный, несмотря на неполное развитие модели, описывающей возникновение явления коллективного порядка в сложных системах, которые начинаются с полного беспорядка.

Самоорганизующиеся карты можно также рассматривать как векторный квантователь, реализуя таким образом принципиальный подход к выводу правила коррекции, используемого для настройки векторов весов [94]. Этот

последний подход еще раз подчеркивает роль функции окрестности, выступающей в роли функции плотности вероятности.

В [71] показано, что динамика обучения самоорганизующейся карты на этапе упорядочивания адаптивного процесса (т.е. во время топологического упорядочивания изначально беспорядочной карты признаков) не может быть описана стохастическим градиентным спуском с одной функцией стоимости. Однако в случае одномерной решетки этот процесс может быть описан с использованием множества функций стоимости - по одной для каждого нейрона сети. При этом эти функции независимо минимизируются в соответствии с методом стохастического градиентного спуска.

В алгоритме SOM удивляет то, насколько он легко реализуем, и при этом довольно сложно анализировать его свойства, используя математический аппарат. Для его анализа некоторыми исследователями был предложен ряд мощных методов, однако их результаты имеют ограниченную область применения. В [72] был приведен обзор результатов, полученных в теоретических аспектах алгоритма SOM. В частности, были подчеркнуты недавние результаты, полученные в [73], [74], где утверждалось, что в случае одномерной решетки можно получить строгое доказательство сходимости алгоритма SOM к единственному состоянию после прохождения им этапа самоорганизации. Этот важный результат был показан для довольно широкого класса функций окрестности. Однако то же самое нельзя утверждать для многомерных решеток.

Было отмечено, что использование алгоритма SOM, позволяет получить некоторый порядок, для анализа признаков. Так самоорганизующиеся карты способны выявить ряд признаков и визуально подтвердить наличие репрезентативных данных при помощи формирования вычислительных карт. При некоторых допущениях было показано, что обучение Хебба ведет к созданию упорядоченных пространственных моделей ориентации с указанием доминирующих признаков, на основе многокритериального анализа.

5. ЭКОНОМИЧЕСКИЙ РАЗДЕЛ

5.1. Выбор и обоснование методики расчёта экономической эффективности

Целью выпускной квалификационной работы является исследование, разработка и программная реализация системы автоматизации процесса прогнозирования основным метеорологических параметров.

Выполнение данной части проекта должно подтвердить экономическую целесообразность разработки, в связи с этим будут рассмотрены вопросы эффективности разработки, реализации и внедрения разрабатываемого модуля.

Основными показателями экономической эффективности являются: срок окупаемости капитальных вложений, экономический эффект, коэффициент экономической эффективности капитальных вложений.

Экономический эффект – выраженный в стоимостной форме результат внедрения программного продукта, исчисляемый в виде экономии от его осуществления. Для организаций, использующих программный продукт, к основным источникам экономии можно отнести: улучшение показателей их основной деятельности, в результате использования программного продукта; уменьшение сроков освоения программных продуктов за счёт улучшенных эргономических характеристик; уменьшение использования машинного времени и других ресурсов на отладку и внедрение ПО в эксплуатацию; повышение уровня технической оснащённости, улучшение качества и уменьшение вычислительных объёмов; сокращение времени получения информации; увеличение коэффициента использования вычислительных ресурсов, средств передачи и подготовки информации; сокращение численности персонала, необходимого для обслуживания программных средств и систем; уменьшение трудоёмкости работ при программировании прикладных задач с использованием лучших программных средств в организации; снижение затрат на эксплуатационные материалы.

Фактический экономический эффект можно рассчитать, сопоставляя затраты и результаты применения программного продукта. Расчёт фактического эффекта учитывает внедрение одного программного продукта на

конкретном объекте, а также общий экономический эффект от использования конкретного программного продукта на всех объектах за определённый расчётный период. Расчёт фактического эффекта необходим для оценки деятельности организаций деятельность которых связана с разработкой, внедрением или использованием программных продуктов, по данному расчёту определяется размер отчислений в фонды экономического стимулирования, а также осуществляется анализ эффективности функционирования программного продукта и вырабатываются технические предложения по его совершенствованию и условиям его эксплуатации.

Для определения ЭППП используются следующие экономические оценки: результативного влияния конечного результата использования ПО (данная оценка является основной для анализа и расчёта показателей эффективности ПО); результат влияния на технологические процессы подготовки, переработки данных, передачи в вычислительных системах (данная оценка является основной для анализа и расчёта показателей для ПО организации вычислительных процессов и эксплуатации средств вычислительной техники, расширяющих функции операционных систем); результатов влияния ПО на технологический процесс создания новых программных изделий (данная оценка является основной для анализа и расчёта показателей эффективности для инструментально–технологических средств разработки и производства ПО).

При необходимости определения ЭППП, входящих в состав автоматизированных систем управления и другие системы через оценку его влияния на конечные результаты функционирования этих систем, доля эффекта от ПО оценивается по коэффициенту его долевого участия в показателях эффективности автоматизированных систем. Расчёт данных показателей осуществляется по результатам основной деятельности организации на основе соответствующих общегосударственных, отраслевых и ведомственных методик. Долевой коэффициент участия ПО в показателях эффективности автоматизированных систем в зависимости от условий расчёта определяются отношением трудозатрат на его разработку к трудозатратам на разработку всей

системы, а также методом экспертных оценок по взаимному соглашению разработчиков, изготовителей и пользователей.

5.2 Расчёт показателей экономической эффективности проекта

Определим расходы на создание модуля и поддержание его в работоспособном состоянии.

1. Определение состава работ и состава исполнителей

Процесс разработки включает в себя следующие этапы:

1. Анализ современных моделей, методов, алгоритмов и программных средств для автоматизации предметной области, выявление основных проблем и осуществление выбора путей их решения.

2. Разработка оригинального метода для определения информационных ресурсов.

3. Разработка модели и алгоритмов управления информационными потоками (по памяти и по времени) между ресурсами компьютерных сетей.

4. Выполнение программной реализации разработанных алгоритмов.

5. Отладка реализованной системы.

6. Проведение испытаний реализованной системы.

Основным документом, регламентирующим вышеперечисленные этапы разработки, является ГОСТ 23501.1–79.

Согласно стандарта, можно выделить следующие стадии: техническое задание – ТЗ (ГОСТ 23501.2–79); технический проект – ТП (ГОСТ 23501.6–80); рабочий проект – РП (ГОСТ 23501.11–81); внедрение – ВП (ГОСТ 23501.15–81).

Распределение перечня работ по стандартизируемым стадиям, показано в табл. 5.1.

Табл. 5.1. Работы по стандартизируемым стадиям

Стадии разработки		Перечень работ	Исполнители		
			Руко- тель проекта	Програм- мист	Администр а-тор сети
Техническое задание					
1	Сбор исходных данных		+	–	–
2	Подбор литературы		–	+	–
3	Анализ аналогичных систем		+	+	–

4	Выбор программных и аппаратных средств	–	+	–
5	Постановка задачи на разработку		+	–
6	Утверждение технического задания	+	–	–
Технический проект				
1	Разработка структурной схемы системы	–	+	–
2	Разработка структуры базы данных МП системы	–	+	–
3	Разработка алгоритмического и программного обеспечения системы	–	+	–
4	Разработка пользовательского интерфейса системы	–	+	–
Рабочий проект				
1	Тестирование	–	–	+
2	Отладка	–	+	–
3	Написание программной документации	–	–	+
Внедрение				
1	Опытная эксплуатация	–	–	+
2	Анализ данных, полученных в результате эксплуатации	+	+	+
3	Корректировка технической документации по результатам испытаний	–	+	+

2. Расчёт трудоёмкости и продолжительности работ

Трудоёмкость выполнения работ по созданию модуля определим, как сумму поэтапной трудоёмкости видов работ, оцениваемых экспертным путём в человеко-днях. Трудоёмкость выполнения работ носит вероятностный характер, так как зависит от множества трудно учитываемых факторов.

Трудоёмкость каждого вида работ определим по формуле:

$$T_i = \frac{3 \cdot T_{min} + 2 \cdot T_{max}}{5}, \quad (5.1)$$

где T_{min} – минимально возможная трудоемкость выполнения отдельного вида работ;

T_{max} – максимально возможная трудоемкость выполнения отдельного вида работ.

Продолжительность каждого вида работ в календарных днях (t_i) в днях, определим по формуле:

$$t_i = \frac{T_i}{W_i} \cdot K_{вых}, \quad (5.2)$$

где t_i – трудоемкость работ, человек–дней;

W_i – численность исполнителей, человек;

$K_{вых}$ – коэффициент, учитывающий выходные и праздничные дни.

$K_{вых}$, рассчитаем по формуле:

$$K_{\text{вых}} = \frac{K_{\text{клд}}}{K_{\text{рд}}}, \quad (5.3)$$

где $K_{\text{клд}}$ — число календарных дней;

$K_{\text{рд}}$ — число рабочих дней.

С учётом количества рабочих дней в 2019 году равного 248, $K_{\text{вых}} = \frac{365}{248} = 1,47$. Поэтапный перечень работ по созданию модуля, с указанием продолжительности, экспертные оценки, расчётные величины их трудоёмкости, рассчитанные согласно формулам (5.1) и (5.2), приведены в табл. 5.2.

Табл. 5.2. Расчёт трудоёмкости и продолжительности работ

Стадии разработки	Трудоёмкость, человеко-дни			Кол-во работников, человек	Продолжительность, календарные дни
	$T_{\min}$	$T_{\max}$	T_i	W_i	t_i
Техническое задание					
1 Сбор исходных данных	2	3	2,4	1	3,6
Окончание табл. 5.2. Расчёт трудоёмкости и продолжительности работ					
2 Подбор литературы	2	3	2,4	1	3,6
3 Анализ аналогичных систем	7	9	7,8	2	5,9
4 Выбор программных и аппаратных средств	2	3	2,4	1	3,6
5 Постановка задачи на разработку	2	3	2,4	2	1,8
6 Утверждение технического задания	3	4	3,4	1	5,1
Технический проект					
1 Разработка структурной схемы системы	5	7	5,8	1	8,7
2 Разработка структуры базы данных МП	12	14	12,8	1	19,2
3 Разработка алгоритмического и программного обеспечения системы	14	19	16	1	24
4 Разработка пользовательского интерфейса системы	14	15	14,4	1	21,6
Рабочий проект					
1 Тестирование	3	5	3,8	1	5,7
2 Отладка	3	4	3,4	1	5,1
3 Написание программной документации	3	5	3,8	1	5,7
Внедрение					
1 Опытная эксплуатация	3	4	3,4	1	5,1
2 Анализ данных, полученных в результате эксплуатации	3	4	3,4	1	5,1
3 Корректировка технической документации по результатам испытаний	3	4	3,4	1	5,1
Общая трудоёмкость разработки			90,4	—	—

Общая трудоёмкость работ составит 90,4 человеко-дней.

3. Построение графика проведения работ

Планирование работ осуществлялось в специализированном пакете Microsoft Project, для визуализации использовалась диаграмма Ганта, показанная на рис. 4.1.

Диаграмма проекта представляет собой таблицу, в которой перечислен поэтапный перечень работ по созданию ПО, а также длительность каждого из этапов. Длительность этапов позволяет рассчитать суммарно длительность всего проекта.

Данные таблицы визуализируются графиком, который отражает последовательность выполнения, продолжительность каждого этапа в виде временных отрезков и зависимость этапов друг от друга.

Наглядность подобного метода планирования облегчает представление последовательности и взаимосвязи отдельных этапов и позволяет осуществлять временной контроль исполнения проекта.

Диаграмма проекта, построенная по данным табл. 5.2, показана на рис. 5.1.

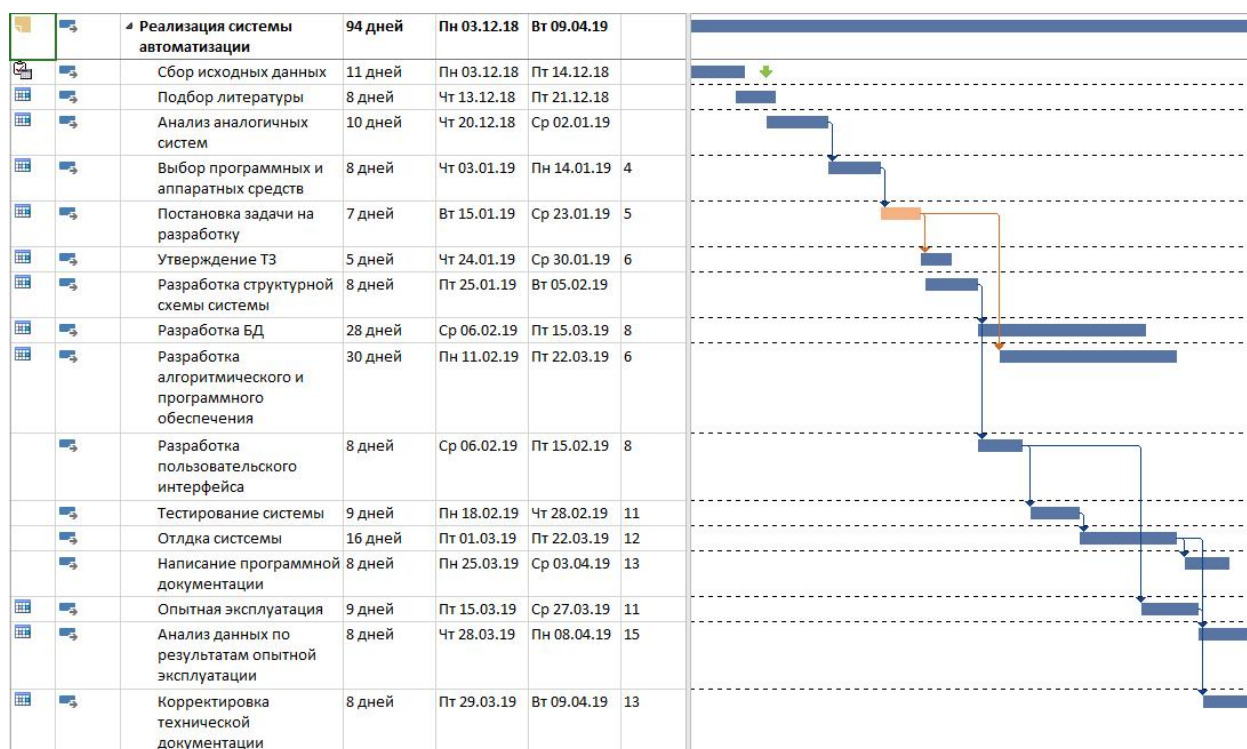


Рис. 5.1. Ленточный график разработки

4. Расчёт сметы затрат на разработку программного продукта

Определим затраты на разработку модуля по следующей формуле:

$$C_{\text{пр}} = C_{\text{осн}} + C_{\text{доп}} + C_{\text{соц}} + C_{\text{м}} + C_{\text{маш.вр}} + C_{\text{н}}, \quad (5.4)$$

где $C_{\text{пр}}$ – стоимость разработки модуля;

$C_{\text{осн}}$ – основная заработная плата исполнителей;

$C_{\text{доп}}$ – дополнительная заработная плата исполнителей, с учётом временных потерь на отпуска и болезни (в среднем 10% от основной заработной платы);

$C_{\text{соц}}$ – отчисления во внебюджетные фонды государственного социального страхования (пенсионный фонд, фонд обязательного медицинского страхования, фонд социального страхования), в среднем составляют 34,2% от основной и дополнительной заработной платы;

$C_{\text{м}}$ – затраты на используемые материалы;

$C_{\text{маш.вр}}$ – стоимость машинного времени.

$C_{\text{н}}$ – расходы накладные включающие затраты на управление, электроэнергию, отопление, ремонт, уборку и др. (в среднем составляют 60% от основной и дополнительной заработной платы).

Основную заработную плату исполнителей $C_{\text{осн}}$, рассчитаем как сумму заработных плат исполнителей (среднедневные значения), по следующей формуле:

$$Z_{\text{исп}} = Z_{\text{ср}} \cdot T, \quad (5.5)$$

где $Z_{\text{исп}}$ – заработная плата исполнителей, среднедневная (руб./дн);

$Z_{\text{ср}}$ – тарифная ставка работника организации разработчика ПО, средняя (руб./чел–дн);

T – трудоёмкость разработки ПО (чел–дн).

В таблице 5.3 показаны затраты согласно статье «Заработная плата».

Табл. 5.3. Основная заработная плата исполнителей

Исполнитель	Оклад, руб./мес.	Оклад, руб./день	Трудоёмкость, чел – дни	Сумма, руб.
Руководитель проекта	32500	1625	16	26000
Программист	21300	1065	57	60705
Администратор сети	12500	625	17	10625
Итого основная заработная плата исполнителей, $C_{\text{осн}}$				97330

Дополнительная заработная плата, устанавливаемая на период разработки ПО, относительно основной составляет 10% от её величины. Рассчитанное значение дополнительная заработная плата равно:

$$C_{\text{доп}} = C_{\text{осн}} \cdot 0,1 = 9733 \text{ (руб.)}$$

Рассмотрим расчёт отчислений во внебюджетные фонды, в том числе: пенсионный, социального страхования; фонды федерального и территориального обязательного медицинского страхования. Отчисления в данные фонды рассчитываются относительно выплаченной заработной платы (основной и дополнительной) и составляют 34% при объёме выплат с начала года менее 463000 руб. на каждого работника. Страхование от несчастных случаев на производстве и от профессиональных заболеваний составляет 0,2% для рассматриваемой 1-й группы риска. Отчисления на социальные нужды рассчитываются относительно выплаченной заработной платы (основной и дополнительной) и составляют 30%:

Рассчитанное значение отчислений на социальные нужды равно:

$$C_{\text{соц}} = (C_{\text{доп}} + C_{\text{осн}}) \cdot 0,3 = 32118,9 \text{ (руб.)}$$

Затраты на используемые материалы включают все затраты на канцелярские товары, бумагу для печатных устройств, носители данных и пр. Данные затраты определяются на основе экспертных оценок. Расчёт расходов на используемые материалы показан в табл. 5.4.

Табл. 5.4. Затраты на материалы

Материалы	Количество, шт.	Стоимость, руб.
Бумага «Снежинка»	5	850
Картридж для принтера	2	5000
Прочие канцтовары	—	1000
Общая стоимость материалов, C_m		6850

Расходы связанные с управлением и организацией работ C_n рассматриваются как накладные расходы и рассчитываются относительно основной заработной платы (в среднем 60% от основной заработной платы исполнителей). Сумму накладных расходов определим по формуле:

$$C_n = C_{\text{осн}} \cdot R, \quad (5.6)$$

где C_n — накладные расходы (руб.);

$C_{\text{осн}}$ — основная заработная плата исполнителей (руб.);

R — коэффициент накладных расходов ($R = 0,6$)

Рассчитанное значение накладных расходов равно:

$$C_n = 97330 \cdot 0,6 = 58398 \text{ (руб.)}$$

Затраты на необходимое машинное время рассчитаем по формуле:

$$C_{\text{маш.вр}} = K_{\text{маш.вр}} \cdot Z_{\text{маш.вр}}, \quad (5.7)$$

где $K_{\text{маш.вр}}$ – стоимость машино-часа, которая включает амортизацию технических средств, затраты на техническое обслуживание и ремонт, стоимость электроэнергии, руб-ч;

$Z_{\text{маш.вр}}$ – время использования технических средств, ч.

Время использования технических средств $Z_{\text{маш.вр}}$ рассчитаем по формуле:

$$Z_{\text{маш.вр}} = T_i \cdot t_{\text{см}} \cdot K_{\text{ср.маш}}, \quad (5.8)$$

где T_i – трудоёмкость работ, чел-дн;

$t_{\text{см}}$ – длительность рабочей смены. При пятидневной рабочей неделе $t_{\text{см}} = 8$ часов;

$K_{\text{ср.маш}}$ – усреднённый коэффициент использования машинного времени примем равным единице.

Значение времени использования технических средств равно:

$$Z_{\text{маш.вр}} = 90,4 \cdot 8 \cdot 1 = 723 \text{ ч.}$$

Стоимость машино-часа, в среднем составляет 15 руб.

Поэтому затраты на использование машинного времени равны:

$$C_{\text{маш.вр}} = 723 \cdot 15 = 10845 \text{ руб.}$$

Расчёта затрат на проектирование ПО приведен в табл. 5.5.

Табл. 5.5. Затраты на ПО

Наименование статей	Обозначение	Сумма, руб.	Процент к итогу
Основная заработная плата	$C_{\text{осн}}$	97330	45,2
Дополнительная заработная плата	$C_{\text{доп}}$	9733	4,5
Отчисления на социальные нужды	$C_{\text{соц}}$	32118,9	14,9
Материалы	$C_{\text{м}}$	6850	3,2
Стоимость машинного времени	$C_{\text{маш.вр}}$	10845	5
Накладные расходы	$C_{\text{н}}$	58398	27,1
Итого:	$C_{\text{пр}}$	215274,9	100

Итого затраты на реализацию ПО составляют 215274,9 рублей. Поскольку система не подлежит тиражированию, расчёт оптовой и розничной цен не производился.

5. Определение трудоёмкости

Исходя из сложившейся на предприятии практики после внедрения ПО для выполнения работ по размещению заказов на ЭТП понадобится 1 человек, использующий 1 компьютер.

Расчёта трудоёмкости по обработке данных на ПЭВМ, в человеко–часах, осуществлялся по следующей формуле:

$$T_i = F_{\text{др}} \cdot W_i, \quad (5.9)$$

где $F_{\text{др}}$ – текущий годовой фонд времени работы работника, ч;

W_i – количество работников, чел.

Трудоёмкость обработки информации с использованием ПЭВМ, при среднем годовом действительном фонде времени работы одного работника за год ($F_{\text{др}}$), который равен 1779 ч, составит:

$$T^6 = 1779 \cdot 1 = 1779 \text{ чел} - \text{ч}.$$

С учётом коэффициента использования ПЭВМ ($L_i = 1$), для реализации ПО, трудоёмкость обработки информации на ПЭВМ, в годовом исчислении, рассчитывается по формуле:

$$T_{\text{обр}} = K_i \cdot L_i, \quad (5.10)$$

где K_i – коэффициент использования компьютера.

Трудоёмкость обработки информации на ПЭВМ с использованием разрабатываемого ПО составит:

$$T_{\text{обр}} = 1779 \cdot 1 = 1779 \text{ маш} - \text{ч}.$$

6. Расчёт капитальных вложений

Суммарная стоимость капитальных вложений в проектом варианте определяется по следующей формуле:

$$K^{\text{пр}} = K_{\text{пом}}^{\text{пр}} + K_{\text{инв}}^{\text{пр}} + C_{\text{разр}}, \quad (5.11)$$

где $K_{\text{пом}}^{\text{пр}}$ – стоимость использования производственного помещения, руб.;

$K_{\text{инв}}^{\text{пр}}$ – стоимость оборудования и инвентаря, руб.;

$C_{\text{разр}}$ – стоимость разработки программы, руб.

Стоимость использования производственных помещений из расчёта 6 кв. м на 1 рабочее место работника, использующего ПЭВМ при стоимости 1 кв. м площади – 115000 руб. составит:

$$K_{\text{пом}}^{\text{пр}} = 1 * 6 * 115000 = 690000 \text{ руб.}$$

Стоимость оборудования и инвентаря (см. табл. 5.6) равна:

$$K_{\text{инв}}^{\text{пр}} = 25200 \text{ руб.}$$

Стоимость разработки программы (см. таблицу 5.5) равна:

$$C_{\text{разр}} = 219771,55 \text{ руб.}$$

Значения капитальных вложения рассчитываются исходя из суммы затрат:

$$K^{\text{пр}} = 690000 + 25200 + 219771,55 = 934971,55 \text{ (руб.)}$$

Табл. 5.6. Стоимость оборудования в проектом варианте

Наименование	Цена за ед., руб.	Кол-во, ед.	Сумма, руб.
Оборудование			
Системный блок: процессор: Intel Core Duo E680 – 2,6 ГГц; память – 8048 Мб DDR3; диск – 1 Тб; видео – 4 Гб; привод – DVD/CD–RW.	100000	1	100000
Монитор Samsung PF 700 PF700M–SF	6500	1	6500
Принтер HP LaserJet P1006	3500	1	3500
Инвентарь			
Компьютерный стол	3200	1	3200
Офисное кресло	1500	1	1500
Канцелярские товары	500	—	500
Итого $K_{\text{инв}}^{\text{пр}}$			252000

7. Расчёт годовых текущих затрат

Расчёт годовых текущих затрат осуществляется по формуле:

$$C_i + C_{\text{м}} + C_{\text{осн}} + C_{\text{доп}} + C_{\text{соц}} + C_{\text{н}} + C_{\text{аморт}} + C_{\text{содерж}}, \quad (5.12)$$

где $C_{\text{м}}$ – сумма затрат на материалы (руб.); $C_{\text{осн}}$ – основная заработная плата исполнителей за год (руб.); $C_{\text{доп}}$ – дополнительная заработная плата работников, с учётом временных потерь на отпуска и болезни (составляет в среднем 10% от основной зарплаты работников) (руб.); $C_{\text{соц}}$ – отчисления на социальные нужды, рассчитываются как 34,2% от основной и дополнительной заработной платы (руб.); $C_{\text{н}}$ – накладные расходы включая затраты на управление, электроэнергию, отопление, уборку, ремонт, и пр. (составляют в среднем 60% от основной и дополнительной заработной платы) (руб.); $C_{\text{аморт}}$ – сумма амортизационных отчислений за год по сопоставленным группам нематериальных активов и основных производственных фондов рассчитывается,

с учётом следующих норм амортизации: от стоимости оборудования – 12,5%; от стоимости производственных площадей – 2,5%; от стоимости программного обеспечения – 30%, принимая расчётный срок эксплуатации ПО за 3 года) (руб.).

$C_{\text{содерж}}$ – затраты на содержание продукта в год (руб.).

Для проектного варианта затраты на материалы составят 700 рублей в год на необходимые канцтовары.

Заработная плата исполнителей (20000 руб. в месяц, для 1 работника) составит:

$$C_{\text{осн}}^{\text{пр}} = 20000 \cdot 12 \cdot 1 = 240000 \text{ (руб.)}$$

Дополнительная заработная плата (составляет от основной заработной платы 10%) составит:

$$C_{\text{доп}}^{\text{пр}} = 240000 \cdot 0,1 = 24000 \text{ (руб.)}$$

Отчисления на социальные нужды (составляет от основной и дополнительной заработной платы 34,2%) составят:

$$C_{\text{соц}}^{\text{пр}} = (240000 + 24000) \cdot 0,342 = 90288 \text{ (руб.)}$$

Накладные расходы (60% от основной и дополнительной заработной платы) составят:

$$C_{\text{сн}}^{\text{пр}} = (240000 + 24000) \cdot 0,6 = 158400 \text{ (руб.)}$$

Амортизационные отчисления складываются из следующих норм амортизации: 12,5% – от стоимости оборудования; 2,5% – от стоимости производственных площадей и 30% – от стоимости программного обеспечения.

Амортизационные отчисления составят:

$$C_{\text{аморт}}^{\text{пр}} = 0,125 \cdot 25200 + 0,025 \cdot 690000 + 0,3 \cdot 219771,55 = 86331,47 \text{ (руб.)}$$

Затраты на содержание продукта в течение года составят:

$$C_{\text{содерж}} = 1779 \cdot 15 = 26685 \text{ (руб.)}$$

Годовые текущие затраты для проектного варианта составят:

$$C_{\text{пр}} = 240000 + 24000 + 90288 + 158400 + 86331,47 + 26685 = 625704,47 \text{ (руб.)}$$

Итак, расходы (выделенные ресурсы) на создание данного модуля и поддержание его в работоспособном состоянии в течение года приблизительно составят:

$C_{\text{общ}} = 625704,47 + 219771,55 \approx 845,5$ (тыс. руб.). За каждый последующий год расходы составят: $C_{\text{пр}} \approx 626$ (тыс. руб.)

5.3. Расчёт величины потерь (рисков), критичных информационных ресурсов после внедрения системы

Согласно данным тестирования системы приведённым в пункте 3.5, рассчитаем величины потерь (рисков) для критичных информационных ресурсов и суммарную стоимость потерь после внедрения, основываясь на мнениях экспертов. Рассчитанные величины приведены в таблице 5.7. Суммарная величина ресурса выделяемого на реализацию системы на первый год, с учётом потерь, после её внедрения составит: $R_{\text{сумм}} = 845,5 + 101 = 946,5$ (тыс. руб.)

Табл. 5.7 Снижение суммарной стоимости потерь после внедрения

Актив	Угроза	Снижение, %	Величина потерь, R_i (тыс. руб.)
Нереализованный прогноз вследствие несвоевременной обработки МД	Высокая	90	40
Нереализованный прогноз вследствие неправильного распознавания МД	Средняя	100	30
Несвоевременный прогноз вследствие низкой скорости обработки МД	Средняя	100	30
Несвоевременный прогноз вследствие неправильного выбора МД	Средняя	100	0
Несвоевременный прогноз вследствие отсутствия специалиста по анализу данных	Низкая	—	1
Суммарная величина потерь после внедрения $R_{\text{прогн}}$			101

Суммарная величина денежных средств выделяемых на эксплуатацию ПО за каждый последующий год, с учётом потерь после внедрения составит: $R_{\text{сумм за посл. годы}} = 626 + 101 = 727$ (тыс. руб.)

8. Оценка динамики величин затрат и потерь

Оценка динамики затрат до и после внедрения ПО, за два года показана в табл. 5.8.

Табл. 5.8. Затраты до и после внедрения ПО, в динамике за 2 года

Состояние	1 кв.	2 кв.	3 кв.	1 год	1 кв.	2 кв.	3 кв.	2 год
До внедрения	280,25	560,5	840,75	1121	1401,25	1681,5	1961,75	2242
После внедрения	236,5	473	709,5	946	1127,75	1309,5	1491,25	1673
Снижение затрат	43,75	87,5	131,25	175	273,5	372	470,5	569

Оценку динамики потерь до и после внедрения, за два года показана в таблице в табл. 5.9.

Табл. 5.9. Потери до и после внедрения ПО, в динамике за 2 года

Состояние	1 кв.	2 кв.	3 кв.	1 год	1 кв.	2 кв.	3 кв.	2 год
До внедрения	280,25	560,5	840,75	1121	1401,25	1681,5	1961,75	2242
После внедрения	25,25	50,5	75,75	101	126,25	151,5	176,75	202
Снижение потерь	255	510	765	1020	1275	1530	1785	2040

Примем допущение о неизменности частоты появления отказов, а также о неизменном уровне надёжности созданной системы.

Тогда, срок окупаемости ПО $T_{ок}$ можно определить аналитическим способом, с использованием следующей формулы:

$$T_{ок} = \frac{R_{сумм}}{R_{ср} - R_{прогн}} \quad (5.13)$$

Срок окупаемости системы составит:

$$T_{ок} = \frac{946,5}{1121 - 101} \approx 0,93 \text{ (года)}$$

Срок окупаемости системы проиллюстрирован на рис. 5.2.

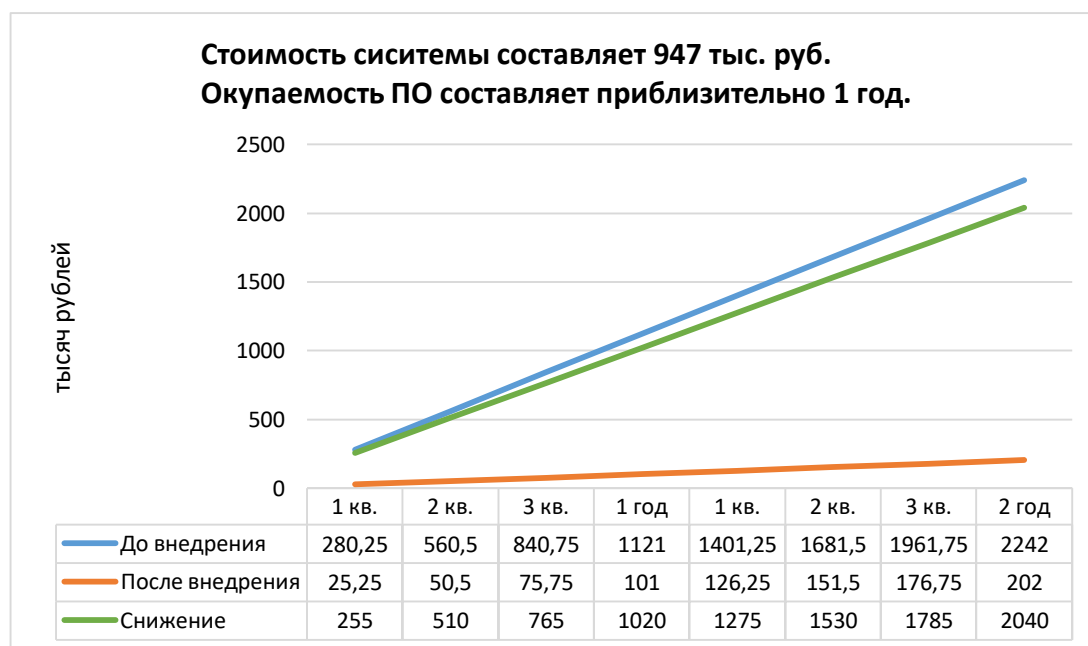


Рис. 5.2. Срок окупаемости системы

Сопоставление сумм потерь (рисков) и сумм, затрачиваемых на реализацию и эксплуатацию системы в течение первого года, показывает экономическую целесообразность разработки и внедрения программы прогнозирования основных метеорологических параметров.