3. Нейросетевой анализ связи между слоями

3.1. Методы нейросетевого анализа связей между слоями

3.1.1. Проблема построения и использования нейросетей в геоинформационных системах

Общая постановка задачи

Опираясь на описание проблемы восстановления пропусков в таблицах, предлагается соединить методы позволяющие делать это наиболее эффективно с программно-инструментальным комплексом, визуализирующим результаты работы. Здесь и далее речь идет об использовании нейросетевых методов обработки информации применительно к географическим информационным системам.

Исходя из предпосылки, что часто даже большое количество информации не может помочь решить проблему, пока она не будет визуализирована на карте, мы приходим к необходимости широкого использования ГИС для обработки и анализа картографических данных. Круг ее возможных потребителей чрезвычайно широк. Прежде всего, это, конечно, управляющие структуры, владеющие большими массивами информации, на основе которых принимаются решения. В картографических данных также нуждаются специалисты, оценивающие и прогнозирующие состояние какой-либо области человеческой деятельности, например, рынков сбыта продукции, загрязнения территории и т.п. Хотя хранящаяся в ГИС информация и представляет собой основную ценность, она приносит практическую пользу только при решении прикладных задач.

В данной работе рассматриваются нейросетевые модели, предназначенные для решения задач относящихся к классу проблем построения функции по конечному набору значений или построение отношений на множестве объектов [35, 78]. К этому классу относятся распространенные и актуальные в ГИС задачи классификации процессов и явлений, районирования и типологии, временной анализ географических комплексов, интерполяция и создание моделей поверхности, анализ и прогнозное картирование пространственно распределенных данных [17, 24]. Такие задачи формализуются как задачи построения действительной функции по конечному набору значений, классификации, анализа временных рядов, выявления зависимостей в данных.

Общая постановка таких задач может быть представлена в следующем виде. Существует набор переменных, описывающих состояние объекта или явления (входных переменных), необходимо найти значения некоторых целевых параметров (выходных переменных). То есть формально. Дано: объект или процесс, который является предметом

исследования. Требуется: Получить значение некоторого зависимого параметра, который характеризует состояние объекта с точки зрения целей исследования.

В [35] подобный тип задач определяется как задача *заполнения пробелов в таблицах данных.* Такое определение предполагает, что постановка задачи может быть представлена в виде таблицы с неизвестными значениями некоторых целевых параметров. Обычно поля таблицы соответствуют выделенным признакам, описывающим объект или процесс, а записи – конкретным примерам проявления этих признаков.

Задача заполнения пробелов в данных в свою очередь порождает задачу выбора метода восстановления. В работе рассматривается нейросетевой метод восстановления информации [11, 35, 48, 76, 79-82].

Для обеспечения эффективного использования нейронных сетей в ГИС необходимо рассмотреть вопросы интеграции и взаимодействия нейросетевых моделей и существующих ГИС.

Проблема построения и использования нейросетевых моделей в ГИС

Проблему применения нейросетевых компонент в ГИС, так же как и в любой прикладной информационной системе можно рассматривать как совокупность следующих проблем:

- Проблема *программной интеграции искусственных нейронных сетей и геоинформационных систем.* Определяет вопросы, связанные с разработкой методов и схем взаимодействия нейросетевых компонент и ГИС, организацией обмена данными и системы запросов между компонентами.
- Проблема *создания нейросетевых моделей в составе геоинформационной системы.* Включает разработку технологии построения нейросетевых моделей, разработку методов автоматизации процесса построения нейросети.
- Проблема *использования нейросетевых моделей в составе геоинформационной системы.* К этой проблеме можно отнести обеспечение устойчивого функционирования, повышение «прозрачности» работы нейросети, получение дополнительной информации о модели, оценку качества работы сети.
- Проблема *технической реализации нейросетевых компонент.* Проблема технической реализации состоит в определении средств построения нейросетевой компоненты, разработке программной системы и обеспечении информационного, программного и технологического соответствия систем.

Проблема интеграции НС и ГИС

ГИС давно уже перестали быть чисто научными инструментами исследователя. Геоинформатика – наука прикладная, решающая проблемы других, тематических областей. Даже самый полнофункциональный ГИС

не может учесть потребности всех и каждого. Реальные же применения имеют свою специфику, которая может сильно расходиться с тем, что предполагал разработчик системы. *Наращиваемая функциональная часть* – одна из важнейших черт современных геоинформационных систем. Можно создать свою, новую функцию, соединяющую сотню уже существующих в ГИС.

Проблема интеграции ИНС и ГИС может быть решена, по крайней мере, тремя способами (рис.3.1):

1) интеграция НС моделей в ГИС;

2) развитие интерфейса между ИНС и ГИС, как самостоятельными системами;

3) создание НС систем с включением интерфейса взаимодействия с ГИС.

Особенности и различия интеграции определяются возможностями ГИС такими как: встроенный язык программирования; средства DDE и OLE; функциональные DLL. Все эти способы требуют написания ГИС-приложения.

ГИС-приложения – специально разработанные для решения каких-то

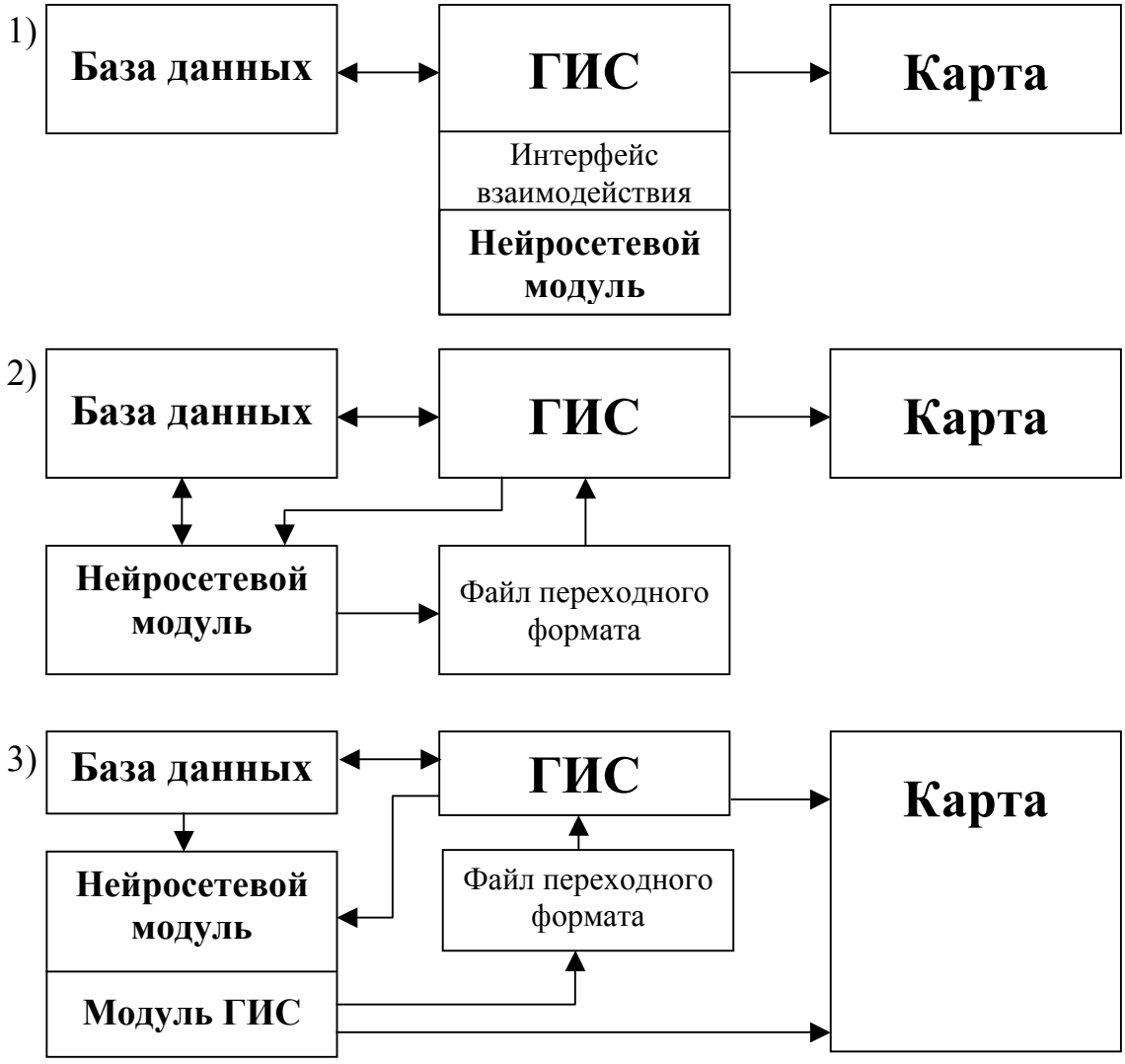


Рис. 3.1. Способы интеграции ИНС и ГИС

конкретных задач алгоритмы обработки данных. Если обобщить известные средства *создания приложений*, их можно четко разделить на две неравные группы.

Первая – преобладающая – располагает собственной, встроенной средой разработки, имеет свой оригинальный язык программирования. Это ArcView, MapInfo, Sinteks. Другая (меньшая) часть ГИС только помогает разработчику создать геоинформационное приложение, а среды разработки в себе не несет. GIS Component (Геоконструктор) GeoGraph –

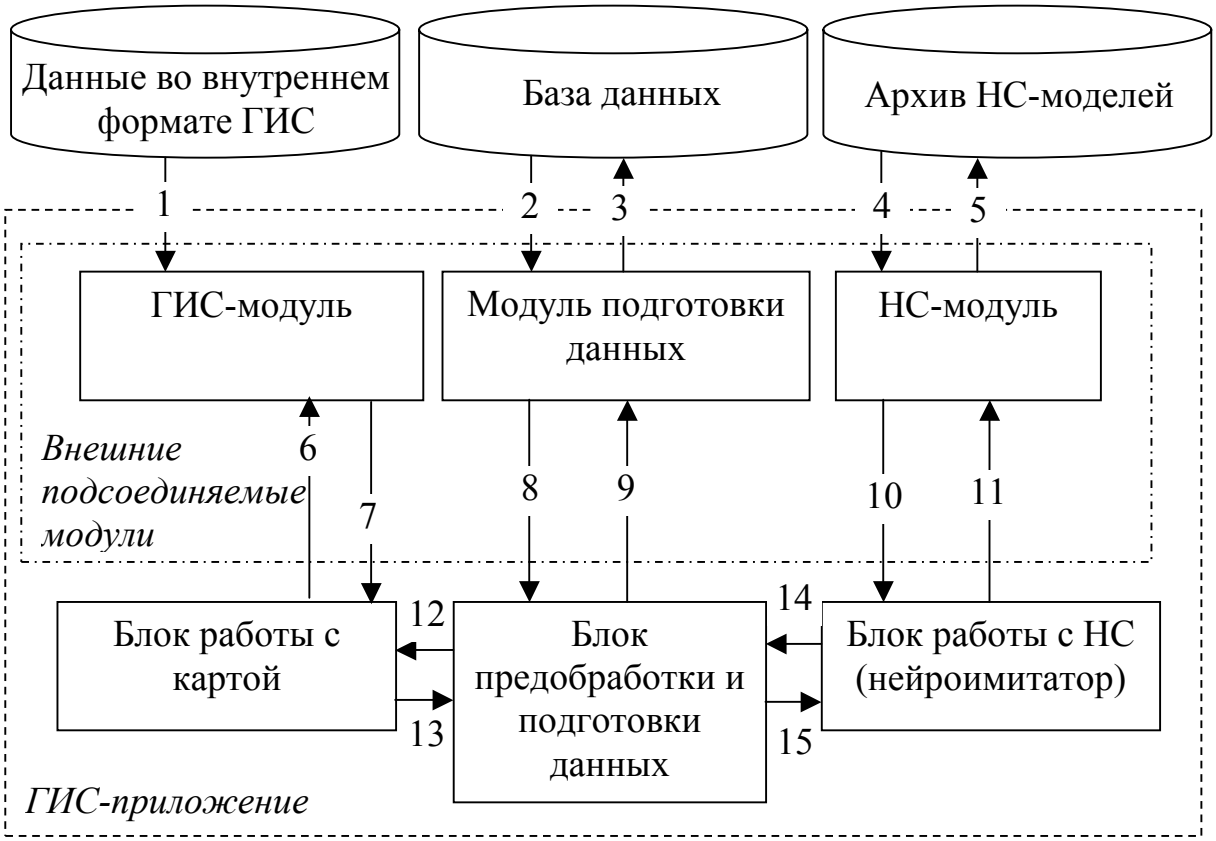


Рис. 3.2. Схема интеграции НС и ГИС.

это библиотека для Visual-сред программирования (Visual C, Delphi); WinGIS работает в режиме DDE-сервера, обслуживая запросы внешних программ. При этом относительно просто достигнуть высокой производительности, программа ориентирована на конкретную архитектуру системы.

В обеих перечисленных группах в последние годы наметились следующие тенденции: переход к объектно-ориентированному методу, ориентация на непрофессионала даже в плане создания приложений, объединение максимума потенциальных возможностей (а именно – обмена с GPS, доступа к "большим" БД, средств импорта из форматов множества других систем) в рамках единой, интегрированной среды.

В работе используется подход, реализующий создание ГИС-приложения на основе библиотеки компонент ГИС (рис.3.2). Программная система созданная по такой технологии может осуществлять стандартный для ГИС набор операций с картой и не имеет ограничения на расширяемость модулей для решения аналитических задач.

Потоки данных между блоками:

1. Файлы карты во внутреннем формате ГИС-модуля.

2-3. Таблицы исходных данных.

4-5. Файлы описания нейросетей.

6. Данные для тематических слоев, запросы к карте, географические операции, манипулирование слоями, задание свойств.

7. Данные о карте, слоях, объектах слоя, связях с таблицами атрибутов.

8-9. Таблицы исходных данных во внутреннем формате системы, параметры и типы предобработки.

10-11. Параметры нейросетей (формирование, обучения, …).

12. Данные для создания тематических слоев.

13. Данные об объектах слоя и связях с таблицами атрибутов (внутренние идентификаторы).

14. Данные обработанные НС. Результаты в терминах поставленной задачи.

15. Предобработанные географические и табличные данные в формате пригодном для обработки нейросетевой моделью.

Интересным также представляется подход интеграции модулей нейросетевой обработки данных в ГИС средствами самой полнофункциональной ГИС. Расчетные модули представляют собой программные динамически подсоединяемые библиотеки DLL. Механизм обмена данными и интерфейс взаимодействия с нейросетевыми моделями разрабатывается с использованием внутреннего языка геоинформационной системы.

3.1.2. Технология построения нейросетевых моделей в составе геоинформационной системы

Одним из эффективных средств решения слабо формализованных задач на основе примеров являются искусственные нейронные сети (ИНС). Распространение нейронных сетей объясняется следующими достоинствами нейросетевого подхода:

1. Автоматическая настройка параметров нейросетевой модели для решения задачи на примерах. Не требуется участие эксперта для построения модели, решающей задачу.

2. Универсальность. Нейросети позволяют стандартным образом, без учета семантики, решать любые задачи, которые допускают представление в виде набора примеров, содержащих входные и выходные данные [35, 83-86].

3. Устойчивость при работе с зашумленными и недостоверными данными [35, 85, 87, 88].

4. Возможность адаптации (дообучения) к новым условиям.

5. Устойчивость к сбоям и разрушениям элементов.

6. Высокий параллелизм, присущий нейросетевым моделям.

7. Способность эффективно обрабатывать данные высокой размерности, разнотипные данные.

В отличие от «традиционных» статистических методов, нейросети выдают не статистически достоверное, а правдоподобное решение задачи и могут применяться при недостатке эмпирических данных для статистического исследования [35]. В качестве достоинств ИНС по сравнению со статистическим подходом можно назвать универсальность и автоматизированный режим настройки в условиях сильной априорной неопределенности, что позволяет быстро получить приемлемый результат. Нейросетевые модели налагают слабые ограничения на возможные функции распределения переменных и позволяют избегать априорных предположений о виде функций распределения переменных и структуре модели [35, 80].

Нейронные сети особенно продуктивны в решении слабоструктурированных задач, так как обычно эксперт может легко структурировать задачу до уровня «черного ящика» или системы данных на основе методов системного анализа [45, 46, 89, 90], то есть указать входные и выходные параметры системы, не указывая метода решения. Как отмечают исследователи, например [35, 77, 79, 91], для решения реальных задач нейросетевой подход является часто более эффективным.

Получение данных из ГИС

Пусть существует набор пространственных данных (сеть мониторинга). Данные представляются в виде: X, Y – пространственные координаты, Z – зависящая от них переменная. Данные, снятые в узлах сетки, нас интересуют по причине того, что такие данные легко представимы в виде двумерной таблицы. Из которой наиболее просто формируется обучающая выборка.

Обычно, первоначально информация существуют в виде сетки мониторинга или отбора проб, поэтому предварительная специальная предобработка не требуется. Если все же данные в ГИС представлены не в сеточном виде (фактически это своеобразный растр) а в векторном то их необходимо предварительно растрировать. Растрировать поля данных можно по-разному. В зависимости от сути поставленной задачи. После растрирования возможно проведение фильтрации для сглаживания, равномеризации или для контраста границ и учета пространственного положения. Алгоритмы получения данных из ГИС описаны в разделе 3.2.1.

В данном разделе остановимся на описании построения нейросетевых моделей для решения задач в ГИС. Основы современной нейроинформатики были заложены в работах [92-94], Теоретические и практические вопросы применения ИНС нашли отражение в работах [35, 36, 75, 78, 80, 85, 95-102]. Наиболее развернуто и последовательно вопрос стандартизации нейросетевых моделей рассмотрен в монографии Е.М. Миркеса [36].

Проект стандарта описан в форме разработки компонент «идеального» нейрокомпьютера. Нейрокомпьютер представлен в виде нескольких модулей, реализующих определенные задачи и взаимодействующих через фиксированный набор запросов. Выделяются следующие компоненты нейрокомпьютера (рис.3.3): задачник, предобработчик, сеть, учитель, интерпретатор, оценка, контрастер, исполнитель.

Остановимся на описании блоков нейросетевой модели необходимых для решения задачи анализа данных в ГИС.

Подготовка обучающей выборки и предобработка данных

При обучении сетей всех видов с использованием любых алгоритмов обучения сети необходимо предъявлять примеры, на которых она обучается решению задачи. Источником данных для сети является задачник.

Формируемый для нейросети задачник представляет собой прямоугольную таблицу, поля которой содержат информацию о входных данных примеров задачи, правильные ответы и другую информацию. Важным вопросом является разделение всех доступных данных на обучающую и тестовую выборки таким образом, чтобы обеспечить их независимость и представительность. Эта проблема решается для каждой конкретной задачи отдельно.

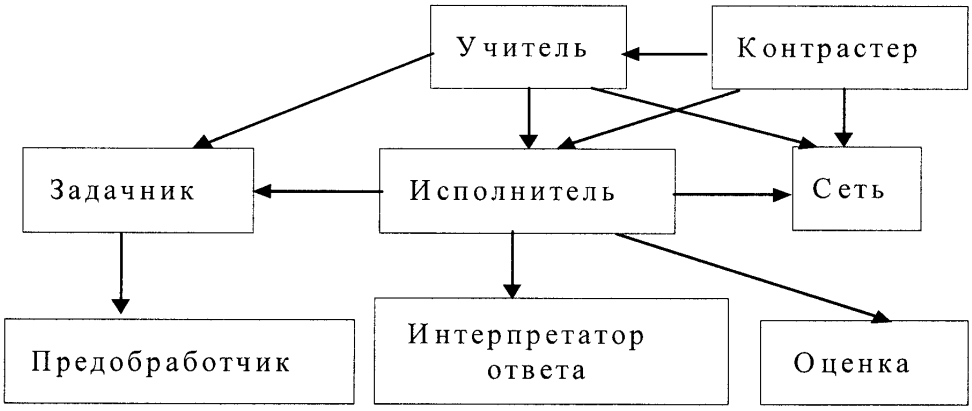


Рис. 3.3. Компоненты нейрокомпьютера

Информация поступает к нейронной сети в виде набора ответов на некоторый список вопросов. Можно выделить три основных типа признаков:

- бинарный признак (возможен только один из ответов – истина или ложь);

- качественный признак (принимает конечное число значений). Для него нельзя ввести осмысленное расстояние между состояниями;
- число.

В процессе обучения параметры подаются на вход сети, значения целевого параметра сравниваются с ее выходом. Однако в большинстве задач использование непосредственных значений из выборки нецелесообразно, т.к. различные переменные имеют различные диапазоны изменения значений, что приводит к затруднениям в процессе настройки весов нейросети. Кроме этого, выборка может содержать как количественные переменные, так и качественные, измеренные в номинальной шкале. Для обеспечения эффективности процесса обучения применяется предварительная обработка данных.

Цель предобработки входных данных – преобразование входных сигналов таким образом, чтобы обеспечить эффективную работу нейронной сети. Для количественных признаков стандартными процедурами предобработки являются нормировка и центрирование, которые обеспечивают универсальность нейронной сети при работе с произвольными данными и позволяют сохранять параметры сети в оптимальном для функционирования диапазоне.

Простейшая предобработка числовых признаков

Числовые сигналы рекомендуется масштабировать и сдвигать так, чтобы весь диапазон значений попадал в диапазон приемлемых входных сигналов. Эта предобработка проста и задается следующей формулой:

$$c' = \frac{(c - c_{min})(b - a)}{(c_{max} - c_{min})} + a, \qquad (3.1)$$

где $[a,b]$ - диапазон приемлемых входных сигналов, $[c_{min}, c_{max}]$ – диапазон значений признака c, c' – предобработанный сигнал, который будет подан на вход сети.

Структура нейронной сети

Нейросети могут рассматриваться как разновидность вычислительных моделей. Возможна общематематическая (или алгоритмическая) запись нейронной сети, отражающая функционирование модели, но нейронная сеть может представлять собой довольно сложную модель, число параметров нейронных сетей, применяемых на практике, может составить порядка 1000 – 10000. «Прямое» описание подобной функции, очевидно, неэффективно. Поэтому для описания сетей выработана специальная «схемотехника», выделены элементы и структуры [35, 36], использование которых позволяет сделать описание нейросети более структурированным, компактным и прозрачным.

Нейронная сеть представляется как конструкция, состоящая из элементарных блоков – синапсов, сумматоров, умножителей, нелинейных элементов и точек ветвления.

На рис.3.4 приведены все элементы, необходимые для построения нейронных сетей. Возможно расширение списка нелинейных преобразователей. Вертикальными стрелками обозначены входы параметров (для синапса – синаптических весов или весов связей), а горизонтальными – входные сигналы элементов. С точки зрения функционирования элементов сети сигналы и входные параметры элементов равнозначны. Различие между этими двумя видами параметров

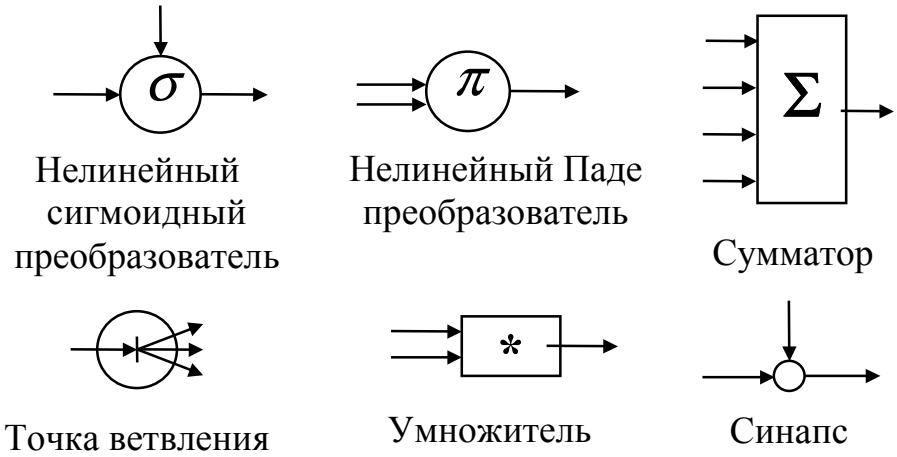


Рис. 3.4. Простейшие элементы сети

относятся к способу их использования в обучении. Также считается, что параметры каждого элемента являются его свойствами и хранятся при нем. Совокупность параметров всех элементов сети называют вектором параметров сети. Совокупность параметров всех синапсов называют вектором обучаемых параметров сети, картой весов связей или синаптической картой. Кроме этого, нужно различать входные сигналы элементов и входные сигналы сети. Они совпадают только для элементов входного слоя сети.

Из приведенных на рис. 3.4 элементов можно построить практически любую нейронную сеть.

Выделим характерные признаки большинства нейросетевых моделей. Прежде всего, нейронные сети являются распределенными структурами и могут быть представлены в виде направленного графа. Узлы представляют процессорные элементы, называемые также нейронами, которые преобразуют информацию в соответствии с некоторой (обычно не сложной) функцией. Ребра задают структуру обмена сигналами между нейронами. Ребрам обычно приписан скалярный параметр – вес связи или синаптический вес. Определены также входные синапсы, по которым подаются компоненты входного сигнала, и выходные, с которых снимается преобразованный сетью сигнал.

В структуре сети можно выделить группы нейронов, функционирующих одновременно и параллельно (слои). В многообразии нейронных сетей можно выделить две базовые архитектуры — слоистые и полносвязные сети. В данной работе используются сети слоистой архитектуры (рис. 3.5).

Нейроны первого слоя получают входные сигналы, преобразуют их и через точки ветвления передают нейронам второго слоя. Далее срабатывает второй слой и т.д. до *k*-го, который выдает выходные сигналы для интерпретатора и пользователя. Если не оговорено противное, то *каждый* выходной сигнал *i*-го слоя подается на вход *всех* нейронов *i* + 1-го. Число нейронов в каждом слое может быть любым и никак заранее не связано с количеством нейронов в других слоях. Стандартный способ подачи входных сигналов: каждый нейрон первого слоя получает все входные сигналы. С такой структурой нейросеть можно представить как сложную вектор-функцию:

$$F^l(\mathbf{a},\mathbf{b},\mathbf{x}) = \sum_{i=1}^{n_p} (a_{i,l}^{p+1} \cdot f_i^p(\mathbf{a}^{p-1},\mathbf{b}^{p-1},\mathbf{f}^{p-1}(...(...,\mathbf{f}^2(\mathbf{a}^1,\mathbf{b}^1,\mathbf{f}^1(\mathbf{a}^0,\mathbf{b}^0,\mathbf{x})))))), \quad (3.2)$$

где l – номер компоненты выходного вектора, **a,b** – векторы параметров или весов связей, **x** – вектор входных данных или переменных, p – число слоев сети, n_p – число нейронов в p-м слое, $f_i^k(\mathbf{a},\mathbf{b},\mathbf{f})$ – функция поведения Паде-нейрона:

$$f_i^{k+1}\left(\mathbf{a}^k,\mathbf{b}^k,\mathbf{f}^k\right) = \frac{\alpha_0^k + \sum_{j=1}^{m_k} \alpha_j^k f_j^k}{c + \beta_0^k + \sum_{j=1}^{m_k} \beta_j^k f_j^k} \quad (3.3)$$

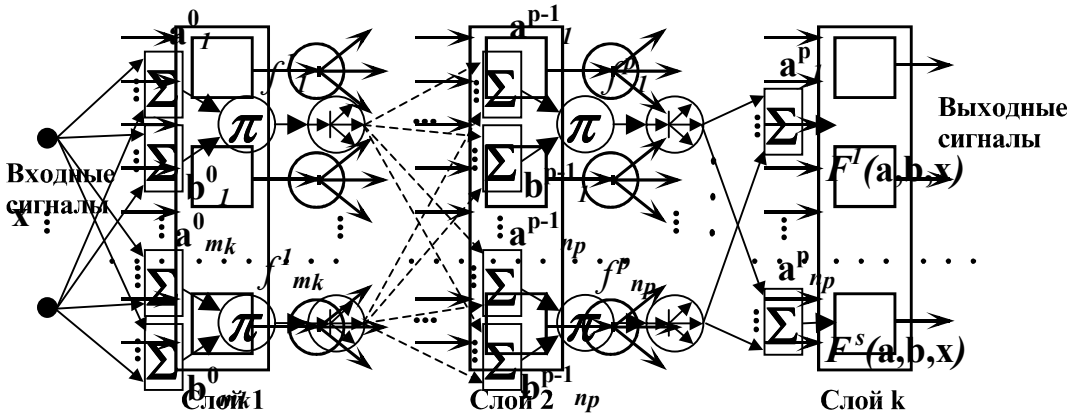


Рис. 3.6. Нейросеть с нелинейными Паде-преобразователями

Рис. 3.5. Сеть слоистой архитектуры

где $c>0$, $k = 1...p$ – номер слоя сети, i – номер нейрона в k-м слое.

Пример нейросети вида (3.2) представлен на рис.3.6.

Функционирование сети

В работе нейронной сети можно различить два процесса: обучения сети и использования обученной сети. При использовании обученной сети происходит только решение сетью определенной задачи. При этом синаптическая карта сети остается неизменной. Работа сети при решении задачи называется ***прямым функционированием***.

При обучении нейронных сетей методом обратного распространения ошибки нейронная сеть (и каждый составляющий ее элемент) должна уметь выполнять обратное функционирование. ***Обратным функционированием*** называется процесс работы сети, когда на *выход* сети подаются определенные сигналы, которые далее распространяются по тем же связям, что и при прямом функционировании до входа сети. При прохождении сигналов обратного функционирования через элемент с обучаемыми параметрами вычисляются поправки к параметрам этого элемента. Если на выход сети с непрерывными элементами подается производная некоторой функции F от выходных сигналов сети, то вычисляемые сетью поправки являются элементами градиента функции F по обучаемым параметрам сети. Исходя из этого требования, определяются правила обратного функционирования для элементов сети.

Покажем это для одного элемента сети: Паде нейрона.

Нелинейный Паде преобразователь

Нелинейный Паде преобразователь или Паде элемент имеет два входных сигнала и один выходной. Обозначим входные сигналы через A, B. Тогда выходной сигнал Паде элемента равен A/B (рис. 3.7).

При обратном функционировании на выход Паде элемента подается сигнал $\partial F/\partial(A/B)$. На входах сигналов A и B должны быть получены сигналы обратного функционирования, равные

$$\frac{\partial F}{\partial A}=\frac{\partial F}{\partial(A/B)}\frac{\partial(A/B)}{\partial A}=\frac{1}{B}\frac{\partial F}{\partial(A/B)} \text{ (рис 3.7а) и} \qquad (3.4)$$

$$\frac{\partial F}{\partial B}=\frac{\partial F}{\partial(A/B)}\frac{\partial(A/B)}{\partial B}=-\frac{A}{B^2}\frac{\partial F}{\partial(A/B)}, \text{ соответственно (рис. 3.7б).} \qquad (3.5)$$

Различные нелинейные преобразователи могут иметь различное количество параметров и различную структуру их описания. Так, например, при n входных сигналах, стандартный адаптивный сумматор имеет $n + 1$ параметр – $y=\alpha_0+\sum_{i=1}^{n}\alpha_i x_i$, Паде-нейрон – $2n + 2$ параметра – $y=\dfrac{\alpha_0+\sum_{i=1}^{n}\alpha_i x_i}{c+\beta_0+\sum_{i=1}^{n}\beta_i x_i}$. Соответственно для сети из k слоев, m нейронов в

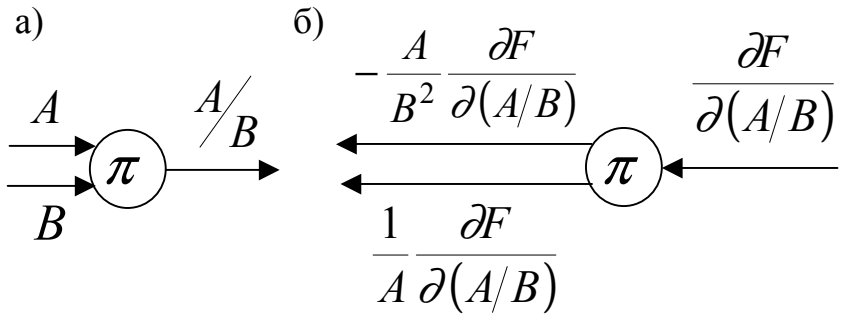


Рис. 3.7. Прямое (а) и обратное (б) функционирование нелинейного Паде преобразователя

каждом слое и *n* входных сигналах количество параметров без учета выходного сумматора равно 2m((n+1)+(m+1)(k-1)).

Исследуем вопрос о степени полиномов в числителе и знаменателе отношения в зависимости от числа слоев в нейросети.

На выходе первого слоя $n^{(1)} = \frac{(a_i, x)}{(b_i, x)}$ имеем отношение полиномов первой степени. На выходе второго слоя

$$n^{(2)} = \frac{(A, n_i^{(1)})}{(B, n_i^{(1)})} = \frac{\sum_i^m A_i \frac{(a_i, x)}{(b_i, x)}}{\sum_i^m B_i \frac{(a_i, x)}{(b_i, x)}} = \frac{\sum_i^m A_i (a_i, x) \prod_{j \neq i} (b_j, x)}{\sum_i^m B_i (a_i, x) \prod_{j \neq i} (b_j, x)} \qquad (3.6)$$

отношение полиномов степени *m*. На выходе третьего слоя полиномы будут степени m^2.

Оптимизационное обучение нейронных сетей

Представить обучение нейронных сетей как задачу оптимизации можно тогда, когда удается *оценить* работу сети. Это означает, что можно указать, хорошо или плохо сеть решает поставленные ей задачи и оценить это количественно. Для этого строится *функция оценки*. Задание функции оценки определенного вида позволяет вычислить для каждого примера обучающей выборки оценку работы нейросети и определить направление изменения ее параметров для уменьшения значения оценки *H*. Процесс итерационного изменения параметров нейросетевой функции называется обучением. Выбор нейросетевого решателя в виде функции (3.2) позволяет применить градиентные методы оптимизации для поиска такого вектора параметров, который доставляет минимум выбранному функционалу оценки.

Функция оценки, как правило, явно зависит от выходных сигналов сети и неявно (через функционирование) – от всех ее параметров. Простейший и самый распространенный пример оценки – сумма квадратов расстояний от выходных сигналов сети до их требуемых значений:

$$H = \frac{1}{S}\sum_{s=1}^{S}\left[\frac{1}{2}\sum_{x\in P_{out}}(Y_s^p(a,b,x_s) - Y^*(x_s))^2\right], \qquad (3.7)$$

где H – оценка работы нейросети, $Y_s^p(a,b,x_s)$ – значение *p-ой* компоненты вектора выходного сигнала нейросети, $Y^*(x)$ – требуемое значение выходного сигнала, S – число примеров.

Устройство, вычисляющее оценку, надстраивается над нейронной сетью, и градиент оценки может быть вычислен с использованием описанного принципа двойственности.

В тех случаях, когда оценка является суммой квадратов ошибок,

$$H = \frac{1}{2}\sum_{x\in P_{out}}(Y(x) - Y^*(x))^2 \qquad (3.8)$$

значения независимых переменных двойственного функционирования $\mu(x)$ для нейронов выходного слоя P_{out} при вычислении градиента устанавливаются равными

$$\mu(x) = \frac{\partial H}{\partial Y(x)} = (Y(x) - Y^*(x)) \qquad (3.9)$$

на вход при обратном функционировании поступают ошибки выходных сигналов. Такой метод вычисления градиента оценки называется методом обратного распространения ошибок.

Переменные обратного функционирования μ появляются как вспомогательные при вычислении производных сложной функции. Описать смысл двойственных переменных и легко представить двойственное функционирование сетей можно через метод неопределенных множителей Лагранжа. Это хорошо описано в [35, 85]. Покажем метод на примере слоистой ИНС с нелинейными Паде преобразователями. Уравнение функционирования Паде-нейрона:

$$x_i^{k+1} = \frac{\sum_{j=1}^{m_{k,i}} \alpha_{i,j}^k x_j^k}{c + \sum_{j=1}^{m_{k,i}} \beta_{i,j}^k x_j^k}; (i=0,...,n_k),(k=0,...,p), \qquad (3.10)$$

где p – количество слоев нейросети, n_k – количество нейронов в k-м слое, $m_{k,i}$ – количество параметров у i-го нейрона k-го слоя. Другие обозначения: x^{p+1} – вектор выходных сигналов нейросети, $\bar{x}$ – вектор ответов, $c > 0$.

Функционирование системы задается набором уравнений:

$$\varphi_i^k = x_i^{k+1} - \frac{\sum_{j=1}^{m_{k,i}} \alpha_{i,j}^k x_j^k}{c + \sum_{j=1}^{m_{k,i}} \beta_{i,j}^k x_j^k} = 0;\ (i=0,...,n),(k=0,...,p), \qquad (3.11)$$

Система (3.11) задает способ вычисления x. Пусть имеется функция $H(\alpha,\beta,x)$. Эта функция зависит от α,β и переменных функционирования x. Как правило, явно функция оценки зависит только от выходных переменных x^{p+1}. $H(\alpha,\beta,x) = H(x^{p+1})$.

Для задачи обучения требуется найти производные $\Delta_{\alpha i}^k = \partial H(\alpha,\beta,x)/\partial\alpha_i^k$ и $\Delta_{\beta i}^k = \partial H(\alpha,\beta,x)/\partial\beta_i^k$. для этого введем новые переменные μ (множители Лагранжа) и производящую функцию W:

$$W(\alpha,\beta,x,\mu) = H(x^{p+1}) + \sum_{k=1}^{p}\sum_{i=1}^{n} \mu_i^k \left(x_i^{k+1} - \frac{\sum_{j=1}^{m_{k,i}} \alpha_{i,j}^k x_j^k}{c + \sum_{j=1}^{m_{k,i}} \beta_{i,j}^k x_j^k} \right); \qquad (3.12)$$

Уравнения (3.11) можно записать как

$$\frac{\partial W}{\partial \mu_i^k} = 0, (i=1,...,n), (k=1,...,p). \qquad (3.13)$$

Заметим, что для тех α,β,x, которые удовлетворяют уравнениям (3.11), при любых μ

$$W(\alpha,\beta,x,\mu) \equiv H(\alpha,\beta,x). \qquad (3.14)$$

Это означает, что для истинных значений переменных функционирования x при данных α,β функция $W(\alpha,\beta,x,\mu)$ совпадает с исследуемой функцией H (функцией оценки).

Подбираем такую зависимость $\mu_i(\alpha,\beta)$, чтобы, используя (3.14), получить для $D_{\alpha_{i,j}^k} = \partial H(\alpha,\beta,x)/\partial\alpha_{i,j}^k$ и $D_{\beta\,i,j}^{\ k} = \partial H(\alpha,\beta,x)/\partial\beta_{i,j}^k$ наиболее простые выражения. Выберем такие μ чтобы:

$$\frac{\partial H(\alpha,\beta,x)}{\partial x_j^k} + \sum_{i=1}^{n} \mu_i^k \frac{\partial \varphi_i^k(\alpha,\beta,x)}{\partial x_j^k} = 0. \qquad (3.15)$$

Исходя из (3.15) находим $\mu_q^k\ (k=1,...,p),(q=1,...,n)$:

$$\mu_q^k = \sum_{i=1}^{n} \mu_i^{k+1} \frac{\alpha_{i,q}^{k+1} - \beta_{i,q}^{k+1} x_i^{k+2}}{c + \sum_{j}^{n} \beta_{i,j}^{k+1} x_j^{k+1}},\ \mu_q^p = -\frac{\partial H(x^{p+1})}{\partial x_q^{p+1}} = \bar{x}_q - x_q^{p+1}. \qquad (3.16)$$

Если μ определены согласно (3.16), то:

$$D_{\alpha_{i,j}^k} = \partial H(\alpha,\beta,x) \Big/ \partial \alpha_{i,j}^k = -\sum_{i=1}^{n} \mu_i^k \left(\frac{x_j^k}{c + \sum_{j=1}^{m_{k,i}} \beta_{i,j}^k x_j^k} \right) \tag{3.17}$$

$$D_{\beta_{i,j}^k} = \partial H(\alpha,\beta,x) \Big/ \partial \beta_{i,j}^k = \sum_{i=1}^{n} \mu_i^k \left(\frac{x_j^k \cdot x_i^{k+1}}{c + \sum_{j=1}^{m_{k,i}} \beta_{i,j}^k x_j^k} \right) \tag{3.18}$$

эти выражения используются для вычисления производных сложной функции вычисляемой нейросетью.

Построение оценки

Если в качестве ответа нейронная сеть должна выдать число, то естественной оценкой является квадрат разности выданного сетью выходного сигнала и правильного ответа. Все остальные оценки для обучения сетей решению таких задач являются модификациями данной.

Приведем пример такой модификации. В [35, 36, 85, 96, 88] отмечается, что процесс обучения нейросети можно значительно ускорить, применяя более специализированную оценку. Пусть при составлении задачника величина $\overline{\alpha}$, являющаяся ответом, измерялась с некоторой точностью ε. Тогда нет смысла требовать от сети обучиться выдавать в качестве ответа именно величину $\overline{\alpha}$. Достаточно, если выданный сетью ответ попадет в интервал $[\overline{\alpha} - \varepsilon, \overline{\alpha} + \varepsilon]$. Оценка, удовлетворяющая этому требованию, имеет вид:

$$H = \begin{cases} 0, & при \left|\alpha - \overline{\alpha}\right| \le \varepsilon, \\ (\alpha - \overline{\alpha} - \varepsilon)^2, & при\ \alpha > \overline{\alpha} + \varepsilon, \\ (\alpha - \overline{\alpha} + \varepsilon)^2, & при\ \alpha < \overline{\alpha} - \varepsilon. \end{cases} \tag{3.19}$$

Эту оценку будем называть оценкой числа с допуском ε.

Оценка вида (3.19) позволяет ускорить процесс обучения и получить более сглаженные нейросетевые функции. В этом случае требования к сложности нейросети становятся более мягкими. Сглаженная функция, в свою очередь, обладает более высокими экстраполяционными и интерполяционными способностями [103, 104].

Для задач классификации также можно пользоваться оценкой типа суммы квадратов отклонений выходных сигналов сети от требуемых ответов. Однако, эта оценка плоха тем, что во-первых, требования при обучении сети не совпадают с требованиями интерпретатора, во-вторых – такая оценка не позволяет оценить уровень уверенности сети в выданном

ответе. Для задач классификации целевая переменная является дискретной (номинальной). Ее кодирование проводится таким образом, чтобы сеть имела несколько выходных полей, каждому из которых соответствует определенный класс. Тогда ответом нейросети на предъявленный пример будет считаться номер класса, соответствующий номеру выходного параметра, на котором зафиксировано наибольшее значение сигнала (интерпретатор «победитель забирает все» [35, 36]). В этом случае в качестве оценки работы нейросети предпочтительнее использовать оценку вида «расстояние до множества правильных ответов», предложенную в [35, 84]:

$$H = \begin{cases} 0, & \forall j \neq k, \quad \alpha_k - \alpha_j \geq \varepsilon \\ \rho, & \exists j \neq k, \quad \alpha_k - \alpha_j < \varepsilon \end{cases}, \tag{3.20}$$

где k – номер «истинного» класса, α^j— выходные сигналы сети, $i=1,...,P$, P – число выходных сигналов, ε - требуемый уровень отличия «истинного» сигнала от остальных, ρ - функция расстояния до множества правильных ответов

$$\rho = \sum_{j=0}^{l} \left(\frac{\sum_{i=0}^{l} \beta_i}{l+1} - \beta_j \right)^2, \tag{3.21}$$

где $\beta_0 = \alpha_k - \varepsilon$, $\beta_j = \alpha_i$ – текущие выходные сигналы (за исключением α_k), переобозначенные таким образом, что $\beta_j > \beta_{j+1}$, $j = 1,...,P - 1$, P – число выходных сигналов, l – минимальный номер, такой, что верно неравенство $\frac{\sum_{i=0}^{l} \beta_i}{l+1} \geq \beta_{l+1}$ при $l < P - 1$, или равенство $l = P - 1$.

Модификация синаптической карты (обучение)

Кроме прямого и обратного функционирования, все элементы должны уметь выполнять еще одну операцию – модификацию параметров. Процедура модификации параметров состоит в добавлении к существующим параметрам вычисленных поправок (для сетей с непрерывно дифференцируемыми элементами вектор поправок является градиентом некоторой функции от выходных сигналов).

На основе формул (3.17) и (3.18) вычисляются производные по всем настраиваемым параметрам сети. Это позволяет использовать градиентные методы оптимизации, например метод наискорейшего спуска. Он заключается в итерационном изменении параметров в соответствии с правилами:

$$\alpha_{i,j}^{k}(t+1)=\alpha_{i,j}^{k}(t)-h\cdot\frac{\partial H(\alpha,\beta,x)}{\partial\alpha_{i,j}^{k}},\ \beta_{i,j}^{k}(t+1)=\beta_{i,j}^{k}(t)-h\cdot\frac{\partial H(\alpha,\beta,x)}{\partial\beta_{i,j}^{k}},\quad(3.22)$$

где t – номер итерации, h – шаг оптимизации. Величина шага вычисляется на каждой итерации путем одномерной оптимизации функции H при фиксированных α и β. В некоторых случаях бывает полезно использовать более сложную процедуру модификации карты.

Во многих работах отмечается, что при описанной выше процедуре модификации параметров происходит неограниченный рост величин параметров. Существует несколько различных методов решения этой проблемы. Наиболее простым является жесткое ограничение величин параметров некоторыми минимальным и максимальным значениями. Очевидно, что для Паде-нейронов нужно ограничивать параметр β в знаменателе. При использовании этого метода процедура модификации параметров β примет следующий вид:

$$\beta_{i,j}^{k}(t+1)=\begin{cases}0, & \beta_{i,j}^{k}(t)+h\cdot D_{\beta_{i,j}^{k}}<0,\\ \beta_{i,j}^{k}(t)+h\cdot D_{\beta_{i,j}^{k}}, & \beta_{i,j}^{k}(t)+h\cdot D_{\beta_{i,j}^{k}}>0,\end{cases}\quad(3.23)$$

Алгоритмы обучения

Все алгоритмы обучения сетей методом обратного распространения ошибки опираются на способность сети вычислять градиент функции ошибки по обучающим параметрам. Таким образом, обучение состоит из вычисления градиента и модификации параметров сети. Обучение двойственных сетей с точки зрения используемого математического аппарата эквивалентно задаче многомерной оптимизации. Однако, существует множество не градиентных методов обучения, таких, как метод покоординатного спуска, метод случайного поиска и целое семейство методов Монте-Карло. Все эти методы могут использоваться при обучении нейронных сетей, хотя, как правило, они менее эффективны, чем градиентные методы.

Изучению градиентных методов обучения нейронных сетей посвящено множество работ [105-108]. Все градиентные методы объединены использованием градиента как основы для вычисления направления спуска.

Наиболее известным среди градиентных методов является ***метод наискорейшего спуска***. Идея этого метода проста: поскольку вектор градиента указывает направление наискорейшего возрастания функции, то минимум следует искать в обратном направлении. Последовательность действий следующая:

1. Вычислить оценку *H2*
2. *H1 = H2*
3. Вычислить градиент
4. Оптимизация шага

5. Модификация параметров
6. Вычислить оценку *H2*
7. Если *H1* - *H2* < Точность то переход к шагу 2

Этот метод работает, как правило, на порядок быстрее методов случайного поиска. Чтобы ускорить обучение и исправить другие недостатки наискорейшего спуска, существует огромное число методов. Например, такие как итерационный и модифицированный партан-методы. Существует также большое семейство ***квазиньютоновских методов***, позволяющих на каждом шаге проводить минимизацию в направлении минимума квадратичной формы. Идея этих методов состоит в том, что функция оценки приближается квадратичной формой. Зная квадратичную форму, можно вычислить ее минимум и проводить оптимизацию шага в направлении этого минимума. Наиболее часто используемые методы из семейства одношаговых квазиньютоновских методов – BFGS метод и метод сопряженных градиентов. Эти методы хорошо зарекомендовали себя при обучении нейронных сетей. Подробно ознакомиться с ними можно в работе [35].

Получение, интерпретация и отображение результатов

Результаты работы нейросети получаются после тестирования. Операция тестирования – это ни что иное, как прямое функционирование сети. При этом на вход сети подаются примеры из всего задачника, а не только обучающее множество. На выходе сети получаем ответы. По ответам можно определить несколько характеристик обученности сети. Если сеть обучена не полностью т.е. решает не все примеры из обучающего множества то можно говорить об ошибке обучения. Соответственно если она нулевая то сеть правильно решает все примеры.

Ошибка обобщения показывает насколько хорошо сеть решает примеры не из обучающей выборки. Иногда бывает полезно не стремиться к нулевой ошибке обучения. В результате можно уменьшить ошибку обобщения. Происходить это может из-за того, что обучающая выборка не репрезентативна т.е. не отражает все особенности данных.

Естественная операция – это интерпретация ответов сети. Этот вопрос подробно описан в [35, 36]. Отметим что для решения задачи предсказания числа обычно обезразмеренные выходные сигналы масштабируют в первоначальный интервал. Для классификации могут быть применены иные методы, например: знаковая, порядковая, нечеткая и другие интерпретации.

В предложенной технологии нейросетевого анализа данных ГИС отображение результата является одной из основных целей обработки данных. После тестирования нейросети данные сохраняются в таблицу атрибутов. Возможно сохранение в ту же таблицу, из которой брались данные для обучающего множества. Например, заполняются пробелы в тех столбцах таблицы, которые служили правильными ответами при обучении нейросети. Кроме этого можно записывать такие характеристики как

относительные значимости входов для получения правильного выхода. То есть насколько каждый входной параметр, относительно других, влияет на построение предсказываемого. Причем такая значимость может быть как общая – целиком для каждого параметра, так и отдельно для каждого примера (каждой точки сетки). Таким образом, при создании тематических слоев получаем поля значимости. Один из примеров того, что дают поля значимости – визуально определяются области для каждого входного слоя, наиболее влияющие на получение выходного.

Рассмотрим формально задачу определения значимости по входным параметрам. Пусть $F(\mathbf{x})$ – нейросетевая аппроксимация некоторой неизвестной функции $y(\mathbf{x})$, после окончания обучения параметры **a** и **b** фиксированы и не влияют на значение $F(\mathbf{a,b,x})$. Описанная нейросетевая парадигма обеспечивает построение непрерывной дифференцируемой функции от **x** и позволяет эффективно вычислять $\frac{\partial F(\mathbf{x})}{\partial x_i}$ в любой точке X. Значения производных могут быть интерпретированы как характеристика влияния входных параметров на значение выходного в данной точке. Если в некоторой точке **x** выполняется условие $\frac{\partial F(\mathbf{x})}{\partial x_i} < 0$ то параметр x_i при увеличении будет уменьшать значение целевого параметра и наоборот. $\left|\frac{\partial F(\mathbf{x})}{\partial x_i}\right|$ – характеризует скорость изменения $F(\mathbf{x})$ при изменении x_i.Оценка по всей выборке может производиться по-разному. Например, сумма модулей, максимум или минимум модуля.

3.1.3. Задачи для нейронных сетей

Приведем методику построения нейросети для решение слабоструктурированных задач анализа зависимостей:

1. Сбор данных, определение входных и целевых параметров, указание их типов (дискретные или непрерывные), диапазонов изменений.
2. Формирование обучающей и тестовой выборки.
3. Выбор вида, структуры, параметров нейросети.
4. Выбор метода оценки, интерпретатора ответов, метода оптимизации и определение их параметров.
5. Формирование нейросети.
6. Определение условий останова работы сети.
7. Обучение нейросети.
8. Определение числа повторений эксперимента с разными начальными условиями (инициализируемой картой синаптических весов нейросети) для сбора статистических данных.
9. Определение критериев выбора лучшей нейросети.
10. Сохранение значений параметров задачника, нейросети и процесса обучения.

11. Исследование модели на оптимальный выбор системы исходных показателей, дублирование исходных признаков, значимость исходных признаков для решения основной задачи. При необходимости – переход к п. 1.

12. Анализ полученных результатов, использование построенной нейросетевой модели.

Теперь коснемся некоторых перечисленных пунктов для конкретных задач в ГИС. Тривиальные и общие решения в некоторых пунктах не затрагиваются. Опишем только особенности связанные с задачами. Большинство модификаций относительно конкретной задачи затрагивает пункт 1. Правильно задав входы и выходы, построив оценку и интерпретатор ответов, облегчающих обучение, сводим решение специфических задач к решению стандартной.

Классификация процессов и явлений

Определение входных и целевых параметров. На вход нейросети подаются исходные данные. На выход, параметры, описывающие классификацию. Нейросеть учиться классификации. После правильного решения всех примеров из обучающего множества сеть считается полностью обученной. При тестировании на тестовой выборке решается задача отнесения к определенному классу объектов, для которых он не был определен. Результаты записываются в таблицу. В ГИС создается тематический слой на основе выполненной классификации.

Районирование, типология

С точки зрения нейросетевой технологии районирование и типология не сильно различаются. Все различие в условии слитности участков территорий, однородных в смысле некоторого критерия или группы критериев для районирования и необязательности выполнения этого условия для типологии. Обе эти задачи, в сущности, сводятся к классификации территориальных комплексов. Однако можно разделить решение в зависимости от условий. Если заданы правила классификации то это обычная предыдущая задача. Если же правила не заданы или, как правило, заданы неявно, то решение такой задачи приобретает смысл решения задачи "классификации без учителя". Изменения затрагивают пункты с первого по седьмой. То есть используется специфические методы создания обучающей выборки и/или вида нейросети и соответственно алгоритмы ее обучения. Например, для решения подобных задач часто используются сеть Кохонена и подобные ей.

Выявление определяющих факторов

Выполняется в пункте 11. Описано в конце раздела 3.1.3.

Временной анализ

Не углубляясь в анализ временных рядов, опишем построение задачника для решения таких задач. В сущности, решается первая задача, то есть классификация. Отличие состоит в формировании обучающей выборки. Если не изменять структуру задачника то просто надо добавить для каждого примера предысторию или вести обучение по скользящему блоку задачника. Например, в качестве примера *N* в момент времени *t* с предысторией глубиной два на вход нейросети подается *N(t-2)+N(t-1)+N(t)*. При обучении нужно требовать предсказание *N(t+1)*. Вопрос определение глубины предыстории в данной работе не рассматривается.

Создание моделей поверхности, интерполяция, анализ и прогнозное картирование пространственно распределенных данных

Создание моделей поверхности и интерполяция сводятся в данном случае к задаче построения функции по конечному набору значений. Нейросетевой метод позволяет строить сложную нелинейную нейросетевую функцию, описывающую данные. При этом на вход нейросети подаются пространственные и атрибутивные показатели, описывающие объект или явление. На выходе требуется получить целевой показатель, характеризующий желаемую поверхность.

Задача анализа и прогнозного картирования состоит из совокупности описанных задач. Например, классификации, предсказания и анализа временных рядов.

3.2. Программные средства и примеры использования

3.2.1. Реализация программного комплекса для нейросетевого анализа данных в ГИС

Особенности разработанной программной системы

Программную систему для нейросетевого анализа данных в ГИС функционально можно разделить на три подсистемы.

- Подсистема взаимодействия с ГИС-компонентой.
- Подсистема получения и подготовки данных.
- Подсистема нейросетевой обработки данных.

1. Подсистема взаимодействия с ГИС-компонентой обеспечивает интерфейс между динамически подсоединяемой библиотекой содержащей программный ГИС-обьект и другими подсистемами. Транслирует запросы системы и вызывает внутренние методы ГИС-объекта, отслеживает события и обеспечивает получение и передачу данных. Позволяет выполнять набор следующих операций:

1) картографические операции:

- Открытие слоев GeoDraw для DOS и GeoDraw для Windows, косметических слоев ГеоГраф 1.5., растровых слоев. Все

сторонние форматы, поддерживаемые ГеоГраф 1.5 (SXF, DX90), могут использоваться только через импорт карт ГеоГраф 1.5 [109].

- Импорт картографических композиций созданных в ГеоГраф 1.1 и ГеоГраф 1.5.
- Функции векторного не топологического редактора для косметических слоев ГеоГраф 1.5.
- Масштабирование карты.
- Варьирование объектным составом карты.
- Создание тематических слоев.
- Оверлейные операции.
- Изменение способа отображения объектов (цвет, тип линии и т.п.), в том числе и определение символогии через значения атрибутов, то есть синхронизация визуализации с изменениями в базах данных;

2) атрибутивные операции:

- Связь таблиц атрибутов со слоями карты.
- Работа с таблицами атрибутивных данных.
- Получение данных об объектах слоя.
- Возможность опрашивать через карту в режиме реального времени базы данных.

Формально состоит из блока работы с картой, блока работы с таблицами и блока "легенды". Физически ГИС-объект встроен в подсистему.

2. Подсистема получения и подготовки данных позволяет получать из карты данные в удобном для нейросетевой обработки виде и производить некоторую предобработку. Технология получения данных из ГИС описано в пункте 3.2.2.

3. Подсистема нейросетевой обработки данных (нейроимитатор) предназначена для построения моделей ИНС и решения с помощью ИНС задач ГИС. Задачи, решаемые нейроимитатором в рамках данной работы:

1) автоматизированное решение задач ГИС на основе нейросетевой парадигмы, описанной в пункте 3.1 третьей главы. Построение нейросетевых моделей, решающих задачи:

- классификации;
- предсказания;
- регрессии;
- прогнозирования значений временных рядов;

2) обеспечение функционирования построенных моделей в составе системы;

3) решение дополнительных информационных задач, в том числе получение дополнительной информации о характере зависимости между входными и выходными параметрами модели. Необходимость решения этой задачи связана с необходимостью получения дополнительной информации при анализе и оценке полученной модели, а также при

планировании принятия решений и исследовании моделируемого объекта или процесса. Это позволяет выявлять информацию о том, как влияет каждый компонент входного вектора нейросети на полученное решение в данной точке пространства входных параметров. На основе этого эксперт может сделать вывод о качестве полученной нейросетевой модели и получить новую информацию об исследуемом объекте при нейросетевой реализации информационной модели.

Структура и функции программной системы

Общая схема взаимодействия подсистем приведена в пункте 3.1.1. третьей главы (рис 3.2). Покажем общие схемы первых двух подсистем и остановимся подробно на нейроимитаторе.

Подсистема взаимодействия с ГИС-компонентой.

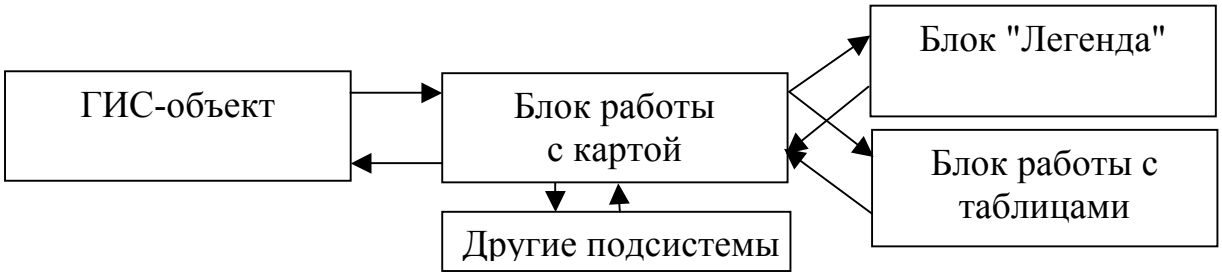


Рис. 3.8. Схема подсистемы взаимодействия с ГИС-компонентой

Подсистема получения и подготовки данных состоит из нескольких блоков (рис 3.9). Предназначена для:

- получения данных из ГИС и преобразования в вид удобный для нейросетевой обработки;
- предобработки полученных данных;
- работы с базами данных;
- обеспечения информационного взаимодействия между подсистемами;
- предварительной визуализации.

Подсистема нейросетевой обработки данных.

Потоки данных между блоками:

1. Таблица исходных данных (выборка).
2. Таблица обработанных данных.
3. Сохранение файлов параметров нейросети.
4. Чтение файлов параметров нейросети.
5. Заданные пользователем или прочитанные параметры процесса обучения.
6. Сохранение параметров обучения.
7. Параметры формирования структуры нейросети.
8. Результаты в терминах поставленной задачи.
9. Данные в формате пригодном для использования блоком обучения. После нормировки, центрирования, кодирования, разбиения на обучающую и тестовую выборки.

10. Созданная (загруженная) нейросеть.
11. Обученная нейросеть.
12. Параметры процесса обучения.

Рис. 3.9. Блоки подсистемы получения и подготовки данных

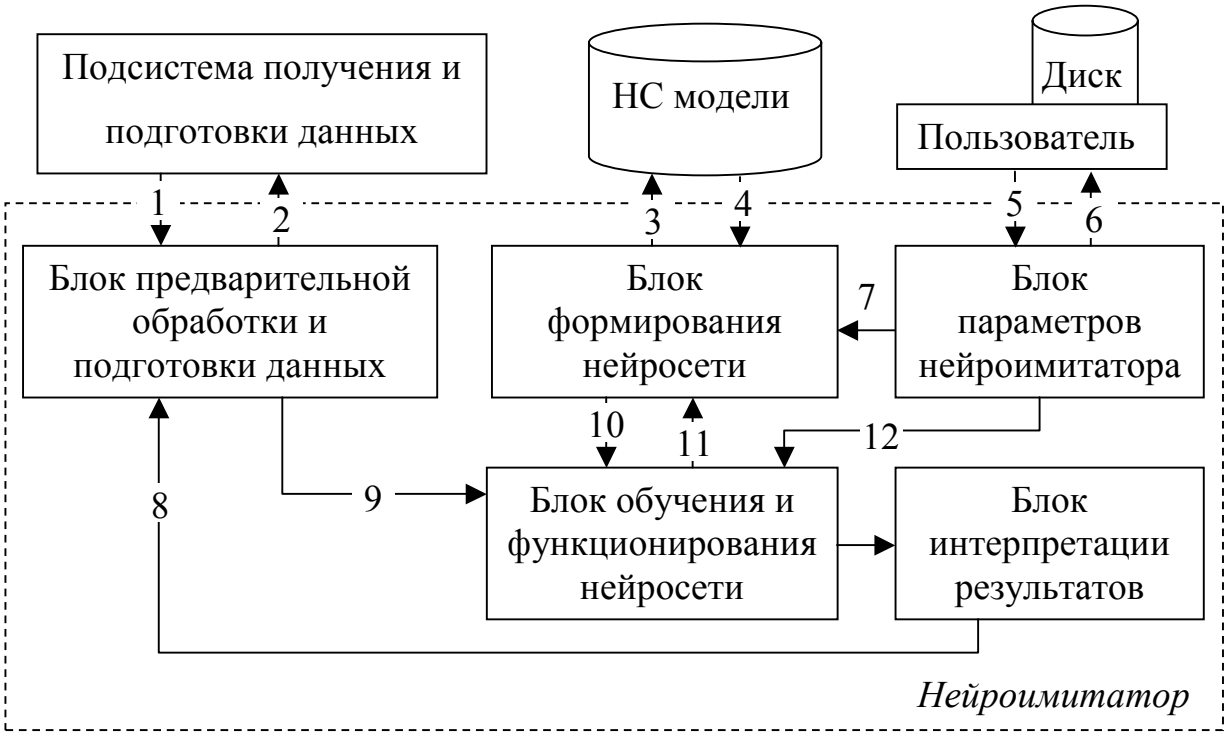


Рис. 3.10. Общая схема структуры нейроимитатора

Подсистема предназначена для непосредственных экспериментов с моделями ИНС и исследования полученных моделей в диалоге с пользователем. Реализованы стандартные функции нейроимитатора: предварительная обработка данных, возможность конструирования структуры многослойной сети, выбор метода оптимизации и вида оценки, обучение и тестирование нейросетевых моделей, определения минимального набора входных параметров, чтение и сохранение нейросетей на диске.

Файл параметров процесса обучения может быть сформирован для каждой задачи, и содержит значения основных параметров обучения, включая настройки формирования структуры нейросети. Файл уже обученной нейросети используется в режиме функционирования системы, а также для дообучения нейросети на новых данных.

Результатом функционирования модуля является файл результатов (таблица), в котором сохраняются значения выходных параметров нейросети, полученные на основе текущих данных. В файле нейросети программа сохраняет полученную нейросетевую модель для дальнейшего использования.

Программная реализация системы

Приведенные выше методы и функции реализованы автором в программной системе GISNNA, предназначенной для решения задач ГИС методами нейроинформатики.

Для реализации программы была выбрана среда программирования Borland Delphi, в состав которой входит компилятор объектно-ориентированного языка высокого уровня Object Pascal и интегрированная среда разработчика. Такой выбор связан со следующими достоинствами Delphi:

1. Среда Delphi предназначена для создания 32-х битных приложений для современной операционной системы Microsoft Windows 95/NT, которая в настоящее время является фактическим стандартом для персональных компьютеров.

2. Объектно-ориентированный язык программирования позволяет сделать процесс разработки ПО более эффективным.

3. Компилятор Delphi строит эффективный исполняемый код, что особенно важно для систем моделирования ИНС, связанного со значительными объемами вычислений.

4. Интегрированная среда разработчика позволяет создавать интерфейс программы с минимальными затратами времени и труда программиста, что повышает эффективность процесса разработки.

Разработанная программная система может быть использована в информационных системах, работающих под управлением ОС MS Windows 95/98/NT и допускающих реляционное представление данных для решения задач.

Основу программной системы составляют модули, описанные в п. 3.2.1.

Взаимодействие с полнофункциональной геоинформационной системой осуществляется на уровне обмена файлами. В программу может быть загружена карта созданная в ГИС Географ. Данные для анализа извлекаются из карты и атрибутивных таблиц. Сохранение результатов анализа производиться в этих таблицах. Запись картографических объектов может производиться в файлы обменных форматов.

Интерфейс программ основан на стандартах GUI (Graphical User Interface) и использует стандартные элементы интерфейса Windows: окна, меню, кнопки, поля ввода и выбора и т.д.

3.2.2. Решение задачи восстановления пропусков

Технология получения данных из ГИС

Конечная цель любой обработки информации в ГИС – это получение картинки (тематического, композиционного, оверлейного и других слоев). Пространственные данные для обработки и для создания тематического слоя электронной карты могут быть получены несколькими способами. Например, по данным дистанционного зондирования или данным изысканий. В большинстве же случаев, пространственные данные вводятся при оцифровке существующих бумажных карт. В результате любого способа ввода пространственной информации образуется слой или слои электронной карты, на которые содержат идентифицированные пространственные объекты, связанные с базами атрибутивных данных.

При вводе пространственной информации как массива координат объектов или непосредственного ввода как точечного объекта электронного точечного слоя карты, как промежуточная задача (иногда как основная) возникает необходимость восстановить непрерывное поле признака по имеющимся дискретным значениям. Иногда наоборот появляется необходимость использовать пространственную информацию с оцифрованных карт изолиний. В явном виде, без аналитического представления данной поверхности, сделать это сложно. Поэтому, для решения задач обработки информации представленной в виде изолиний (в силу специфичности обработки), приходится решать обратную задачу восстановления дискретных значений и построения поверхности по полученным точкам.

Для нейросетевого анализа данных в ГИС удобно использовать данные в виде описанном в разделе 2.3 второй главы. То есть в виде данных на сетке или матричном виде. Это означает что в узлах сетки конкретным координатам этих узлов соответствуют некоторые количественные и качественные значения явлений (или различные показатели одного явления). Такая пространственная информация, выраженная в цифровой форме и взятая в одних и тех же точках со всех

исходных карт, позволяет строить пространственно – цифровые модели, проводить статистические и другие необходимые расчеты, проводить нейросетевую и не только обработку данных для поиска взаимосвязей, анализа рассматриваемых явлений, проведения численных экспериментов. В результате таких математических операций в узловых точках создаются "каркасы" из полученных данных, на основе которых можно создавать новые расчетные карты.

Если данные изначально представлены в сеточном (растровом) виде то можно перейти к сразу к задачам обработки и анализа. Если нет то получение данных требует описания. Задачу получения данных из ГИС в приемлемой для обработки форме может быть разбита на несколько подзадач.

- Создание сеточного слоя для растрирования векторного.
- Получение матричной модели слоя по векторным данным в узлах регулярной сетки цифровой поверхности.
- Постобработка. Различного вида фильтрация (сглаживание, выделение контуров, контраст …)
- Визуализация матричных моделей, ввод и тематическая доводка слоя в ГИС.

Опишем технологию поэтапно. Пусть есть карта представляющая собой композицию тематических векторных слоев описывающих разные аспекты (явлений, временные…) одной территории.

Сначала программно создается сеточный слой. Слой задается тремя типами объектов: полигонами – квадратики, линиями – границы квадратов сетки и точками – центры полигонов. Масштаб детализации задается размерами ячеек сетки, т. е. количеством узлов. Не имеет смысла строить поверхность с точностью превышающей точность исходных данных, другими словами, размер ячейки должен быть не меньше среднего шага цифрования (линейных отрезков, которыми выполнялась векторизация), зависящего от масштаба исходной карты. Затем слой добавляется в композицию тематических слоев средствами ГИС.

Второй этап состоит в получении данных в каждой точке сетки из каждого слоя. Чтобы найти значение в каком-либо слое в определенной точке, нужно решить задачу принадлежности точки полигону растрируемого слоя. Затем найти в таблице атрибутов по идентификатору полигона его значение и присвоить его точке сетки. В результате получается прямоугольная таблица. Столбцы – признаки (координаты X и Y, значения слоев), строки – точки. Из этой таблицы далее будет образован задачник для обучения нейросетей.

Выполнение третьего этапа не обязательно и продиктовано особенностями векторного представления полей признаков. При векторизации поля данных огрубляются путем определения линий уровня с некоторым шагом и создания полигонов на их основе. После растеризации полигональных слоев ситуация в точности полей

кардинально не меняется. Но иногда бывает полезно сгладить данные и убрать крупные линейные участки (рис. 3.11).

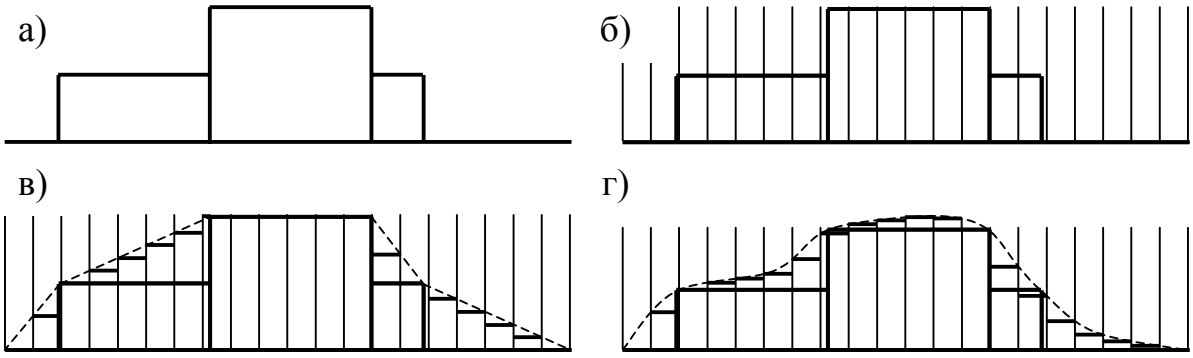
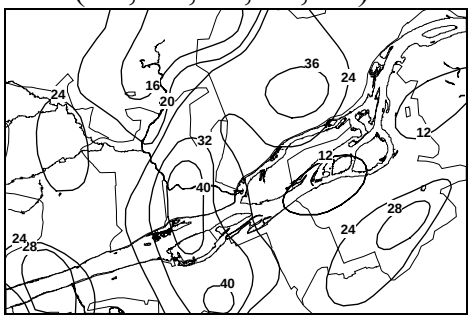
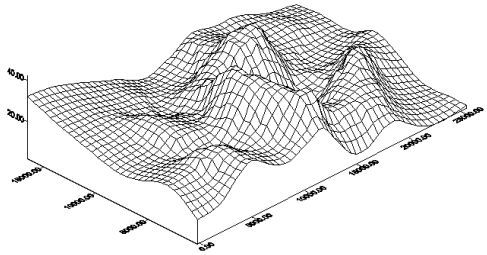


Рис. 3.11. Профильное сечение слоя: а) полигональный слой, б) растрированный по сетке, в) сглаженный кусочно-линейно, г) сглаженный фильтром

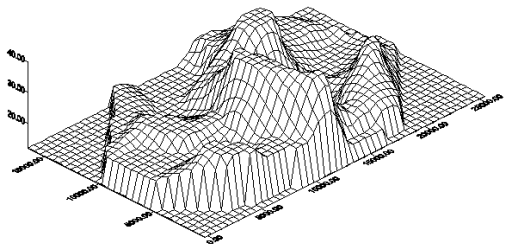
Существует большое семейство методов направленных на восстановление непрерывных полей признаков по дискретным данным и получения в дальнейшем значений в узлах регулярной сетке. Наиболее известные методы: минимальной кривизны, триангуляции, ближайшего соседа, кригинга, радиальных базисных функций, обратного расстояния в степени. Для этих же целей можно использовать и нейросетевые модели [17, 110].

Для примера рассмотрим построение поверхности по данным изолинейного слоя "Суммарное содержание веществ с канцерогенными свойствами в почве" из электронного экологического атласа города Красноярска [111-114] (рис. 3.12). Суммарное содержание канцерогенных веществ (As, Be, Cr, Ni, Pb) в почве. Изолинии условных содержаний.

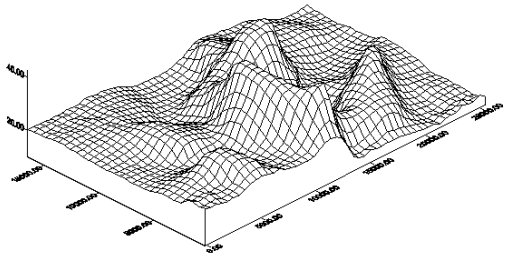




а) минимальной кривизны



б) триангуляции



в) кригинга

Рис. 3.12. Поверхности построенные различными методами

Для визуализации данных, представленных в виде матрицы узловых значений сетки, необходимо подготовить полигональный слой, состоящий из полигональных объектов в виде квадратиков – пикселов. Размеры

пикселов берутся в соответствии с размерами используемой сетки. Таблица узловых значений сетки, которую необходимо визуализировать, сохраняется в качестве файла базы данных, читаемых ГИС – программой. Такая база данных загружается в качестве атрибутивной информации созданного полигонально-пиксельного слоя. После идентификации объектов слоя каждому пикселу будет присвоено значение определенного узла в качестве атрибутивной информации. В дальнейшем определяется шкала тематической раскраски и в готовом слое электронной карты с помощью цвета пикселов отображается пространственное изменение расчетного признака или явления по данной территории (рис. 3.13).

Технология восстановления пропущенных данных

Для формирования и настройки нейросетевых моделей использовалась технология и методы построения нейросетей, разработанные во второй главе данной работы.

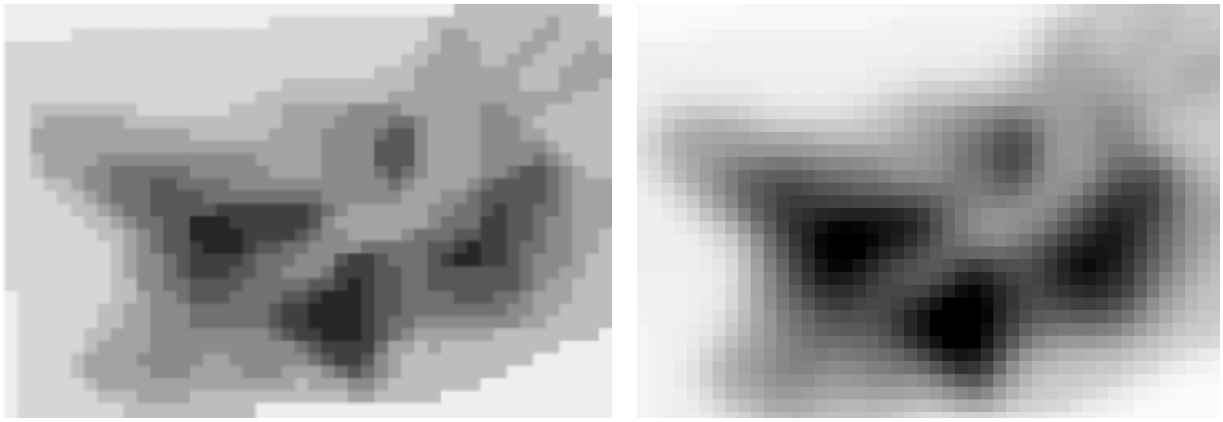


Рис. 3.13. Растрированный и затем сглаженный слой

Перейдем к основной части нейросетевой обработки данных в ГИС. Напомним, что основная задача, которая решается, это задача восстановления пропущенных данных то есть заполнения пробелов в таблице. Создание таблицы описано в разделе 3.2.1. Следующий шаг – это создание задачника для нейросети.

Задачник (обучающее множество) состоит из примеров вида "вход– правильный выход" [18, 35, 36, 80, 115]. Вход – вектор сигналов, предъявляемый нейросети, выход – вектор сигналов, который нейросеть должна выдать после функционирования. Затем производиться нормировка. Например, непрерывные признаки нормируются согласно (3.1). Для получения ответов в реальном масштабе измерения целевой переменной производится обратное преобразование выходных сигналов сети.

По паре "требуемый выход – полученный выход" строиться функция оценки (3.7).

Формируется нейросеть, вычисляющая функцию (3.2). Она состоит из нелинейных преобразователей сигнала, связанных между собой посредством адаптивных связей, которые линейно преобразующих

проходящий через них сигнал (рис. 3.5). Такие элементы называются нейронами. В нашем случае нейросеть состоит из Паде–нейронов (рис. 3.6).

Паде–нейрон состоит из двух адаптивных сумматоров и нелинейного элемента с двумя входами, вычисляющего частное от своих входных сигналов согласно (3.3) (рис. 3.7). Название происходит от аппроксимации Паде – метода рациональной аппроксимации функций.

С выбранным количеством слоев и нейронов в слое создается слоистая нейросеть. Начальные значения весовых коэффициентов адаптивных сумматоров нейронов инициализируются случайными величинами, распределенными равномерно в интервале [-0.01, 0.01]. После выбора метода обучения (см. гл.3. п. "Алгоритмы обучения") нейросеть запускается на обучение. Обучением называется процесс минимизации функции оценки. Он выполняется с привлечением градиентных методов оптимизации и состоит в поиске таких значений параметров, при которых нейросеть выдает правильный вектор выходных сигналов. (см. гл.3. п. "Оптимизационное обучение нейронных сетей")

В качестве метода оптимизации используется один из трех методов: модифицированный Partan, метод сопряженных градиентов, BFGS метод. В качестве функции ошибки используются модификации метода наименьших квадратов «МНК с люфтом» для задач регрессии и «расстояние до множества правильных ответов» для задач классификации. Останов процесса обучения происходит при выполнении одного из условий: 1) ошибка сети на обучающей выборке, полученная с помощью выбранного метода оценки, не превышает заданный уровень; 2) уменьшение ошибки сети не превышает заданного значения. Нейросеть считается обученной после достижения заданного (малого) значения функции оценки то есть при выполнении первого условия останова. Градиентные методы поиска приводят, в общем случае, к локальным экстремумам, которые часто не обеспечивают требуемого качества обучения. Применена следующая эвристическая схема борьбы с локальными минимумами. В случае прекращения уменьшения ошибки сети в процессе обучения производится добавление равномерно распределенной случайной величины к весам связей сети (процедура "удар") и обучение продолжается. Если в результате добавления не удается существенно уменьшить оценку, увеличивается интервал значений случайной величины и "удар" повторяется. Ширина интервала значений последовательно расширяется от значения 0.02 до значения 0.3 с шагом 0.02. Если в результате применения наибольшего по величине сдвига и последующего обучения оценка сети не уменьшилась, процесс обучения сети данной структуры прекращается.

Встречаются ситуации, когда нейросеть плохо обучается из-за противоречивой обучающей выборки и увеличение размера нейронной сети мало помогает. Это бывает когда в обучающей выборке, присутствуют задачи с одинаковыми условиями, но разными ответами

(одинаковыми входными векторами данных, но разными выходными). Появление таких конфликтных примеров может, например, означать недостаточность набора входных признаков, поскольку при расширении признакового пространства конфликтным примерам могут соответствовать разные значения добавляемого признака и критическая ситуация будет исчерпана. В любом случае пользователь должен решить эту проблему, например, исключением конфликтных примеров из задачника.

После построения, нейросетевая модель решения сохраняется на диске, и далее в процессе функционирования ГИС-приложения загружается для решения задач. Адекватность и качество функционирования модели оценивается пользователем или экспертом. При необходимости, строится новая модель или производится настройка имеющейся на основе новых данных (дообучение).

В процессе использования программы, пользователь может дополнительно получить информацию о влиянии компонент входного вектора на значение целевого параметра в заданной точке.

В связи с тем, что нейронные сети являются вероятностными моделями, может существовать несколько различных нейросетей, решающих одну и ту же задачу. При этом результаты решения дополнительных информационных задач, полученные на основе этих нейросетей, могут быть различны. Поэтому для получения более точного представления о процессе получения значений целевого параметра может быть полезным использование нескольких нейросетевых моделей.

После обучения нейронной сети необходимо провести ее тестирование на тестовой выборке для определения точности решения не входивших в обучающую выборку задач. Точность правильного решения очень сильно зависит от репрезентативности обучающей выборки. Обычно при решении различных неформализованных задач в разных проблемных областях точность в 70-90% правильных ответов на тестовой выборке соответствует проценту правильных ответов при решении этих же задач специалистом-экспертом.

Для моделирования искусственных нейронных сетей использовался разработанный программный комплекс. Программа позволяет выполнять как элементарные операции с картой, так и традиционные операции над нейронными сетями (создание сети, ее обучение, тестирование), а также предобработку (растрирование, фильтрация) и визуализацию результатов. Вычислительные эксперименты проводились на персональных компьютерах.

Целью экспериментального использования разработанных методов была проверка работоспособности и экспериментальное изучение возможностей предложенных методов и программных средств.

В качестве примеров решения практических задач с использованием разработанных средств рассматривается задача в области анализа медико-экологогических данных [116-119].

В работе использовалась электронная экологическая карта города Красноярска [112, 113]. Карта содержит данные о морфоструктуре рельефа, плотности и фитопатологическом состоянии растительного покрова в городе, загрязнениях некоторыми, в том числе и канцерогенными, веществами почв и снежного покрова. А также информацию о населении города. Это плотность жилищно-промышленной застройки, динамика показателя заболеваемости злокачественными новообразованиями жителей Красноярска.

Исходя из предположения о пространственной связи повышенных загрязнений атмосферы и почвы с ростом заболеваемости была поставлена задача заполнения пробелов в данных слоя заболеваемости на основе данных из других слоев [120-128].

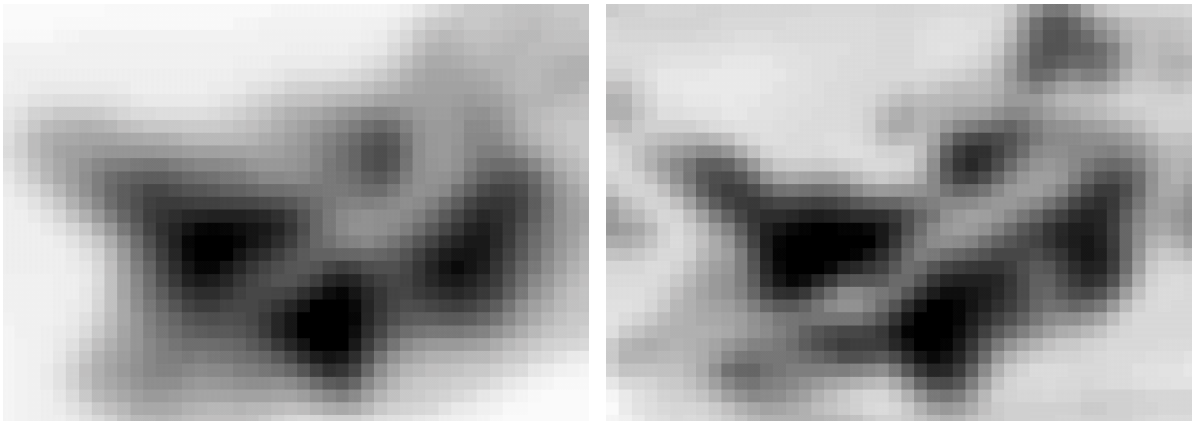


Рис. 3.14. Исходный слой и слой по результатам работы нейросети через несколько циклов обучения

Эксперимент проводился два раза. Каждый раз было обучено по десять нейросетей. Первый раз на вход подавались данные, характеризующие экологическое состояние города, на выходе нужно было получить количество заболевших. Во второй раз к исходным данным добавлялись координаты X, Y.

Обучающее множество создавалось по половине известной информации. Тестирование проводилось для всего слоя. Ошибка обучения была сведена к нулю. Ошибка обобщения составляла 10 – 20 %.

Замечено что нейросеть за несколько первых циклов обучения воссоздает поверхность качественном смысле и все остальное время обучения приближает ее до точных числовых значений (рис. 3.14).

Обученная нейросеть может распространять знания о зависимости между слоями на отсутствующие области карты. При тестировании нейросети по задачнику и примерам, не входящим в задачник, результат записывается в таблицу, после чего, полигональный слой ("сетка") может быть раскрашен в соответствии с полученными при тестировании значениями.

Исследовался также вопрос значимости исходных показателей для решения задачи (рис. 3.15–3.16). Какие из входных сигналов являются доминирующими, (значимыми) при принятии нейросетью решения, а

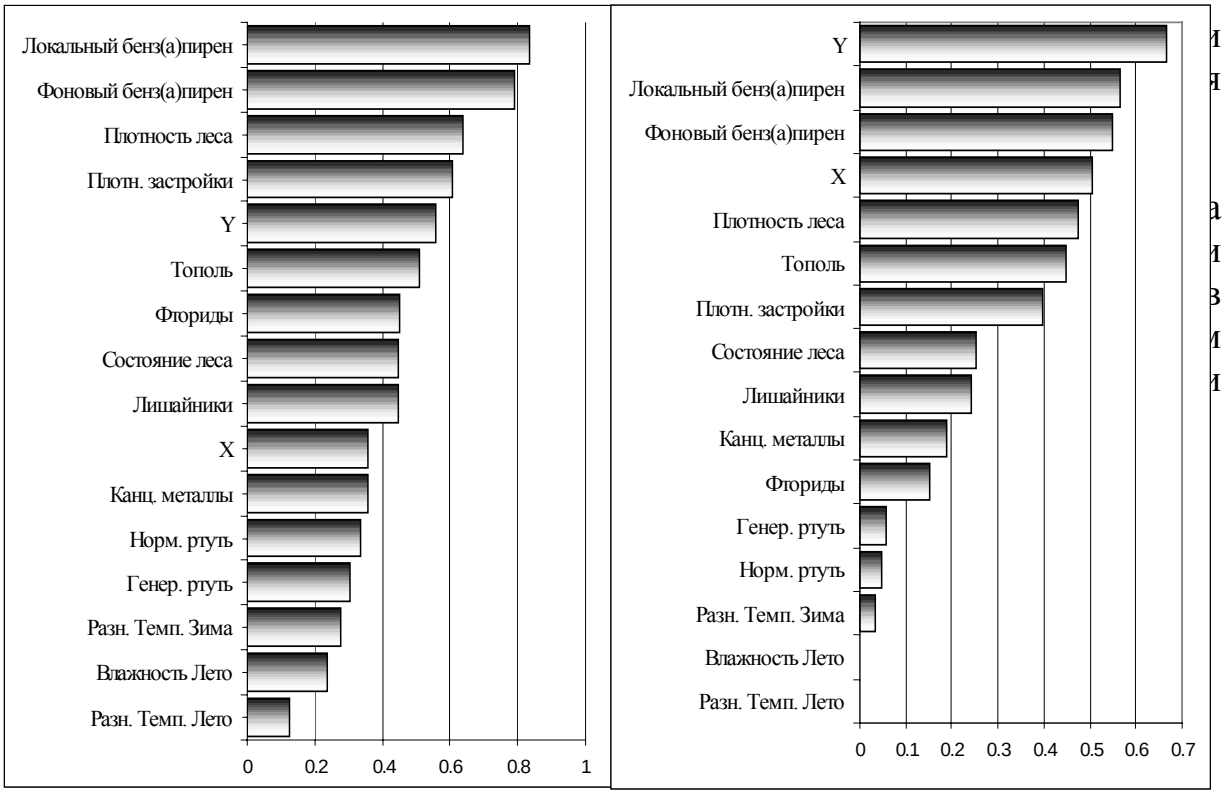


Рис. 3.15. Распределение относительной значимости по входным параметрам с участием координат. Левый график первоначальный. Правый после сокращения незначимых признаков.

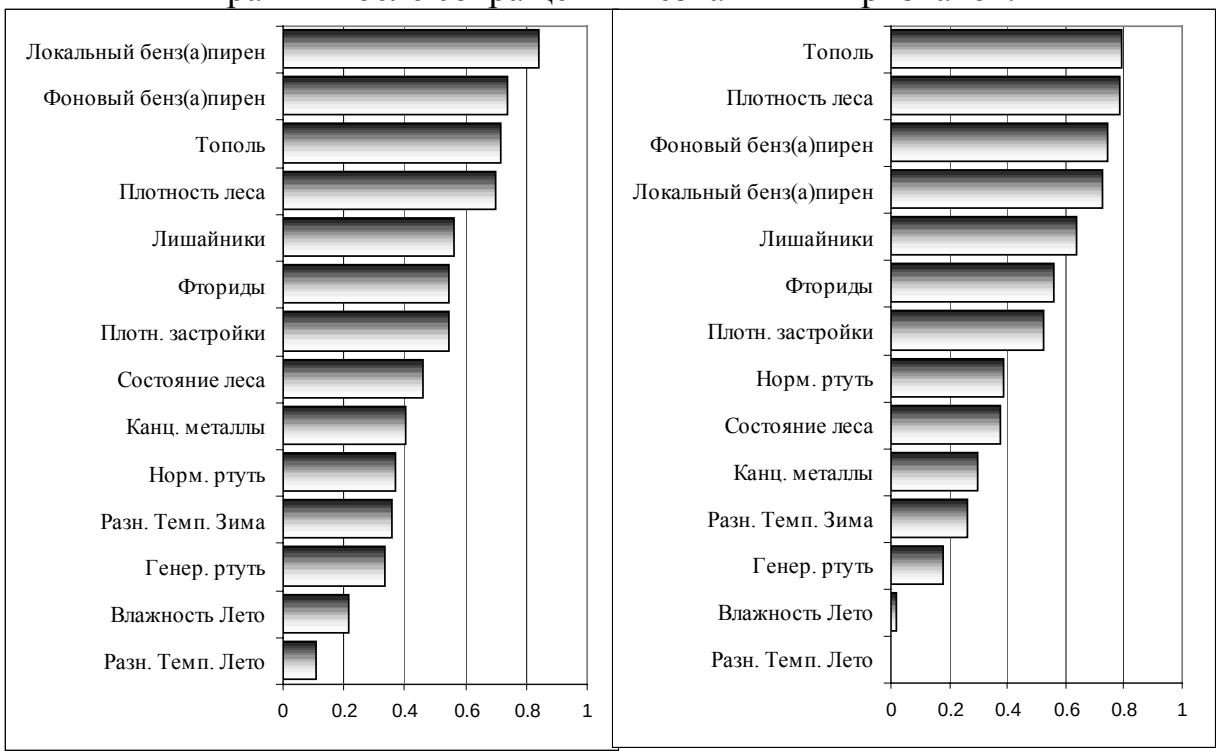
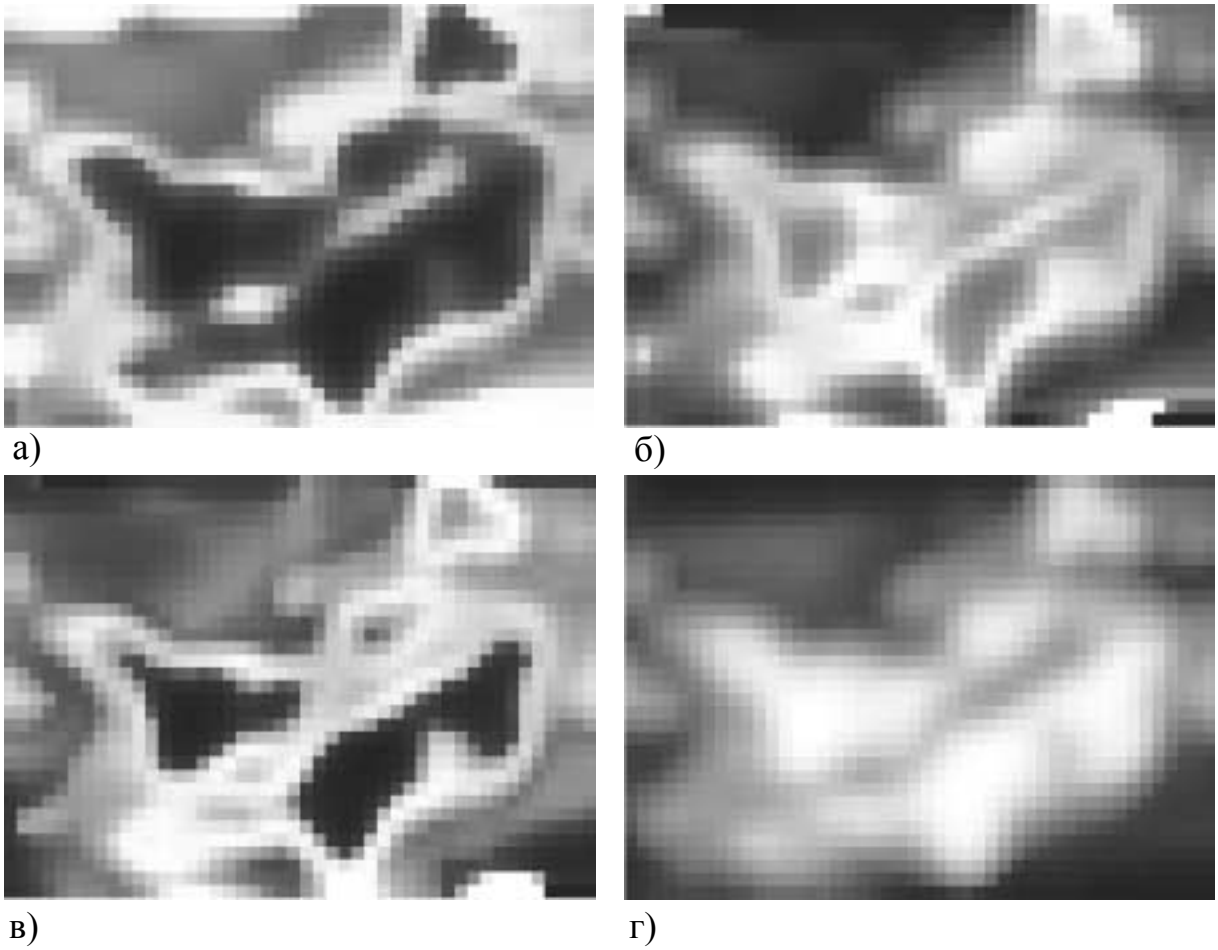


Рис. 3.16. Распределение относительной значимости по входным параметрам без участия координат. Левый график первоначальный. Правый после сокращения незначимых признаков.

Поля значимости рассчитываются как частные производные нейросетевой функции от входных параметров (см. гл. 3.1. п. "Получение, интерпретация и отображение результатов"). Значения производных могут быть интерпретированы как характеристика влияния входных параметров на значение выходного в данной точке. Если в некоторой точке значение производной по параметру меньше нуля то при его увеличении будет уменьшаться значение целевого параметра и наоборот. На рис. 3.17 а), б) и в) светлые области соответствуют отрицательным значениям производных темные соответственно положительным. Серый цвет примерно соответствует нулевым изменениям. На рис. 3.17 г) все значения производных положительные. Белые области близки к нулевым значениям.



а) б)

в) г)

Рис. 3.17. Пример распределения значимости по входным слоям