

А. Л. Серебренников

**СРАВНИТЕЛЬНЫЙ АНАЛИЗ НЕЙРОСЕТЕВЫХ ПАКЕТОВ
И МЕСТО СРЕДЫ SIGNIFICO СРЕДИ НИХ.
КРАТКОЕ ОПИСАНИЕ СРЕДЫ**

**1. ОБЩИЕ СВЕДЕНИЯ О ПРИЛОЖЕНИИ НЕЙРОСЕТЕВОГО
МОДЕЛИРОВАНИЯ**

Программу моделирования нейронной сети обычно называют нейропакетом, понимая под этим программную оболочку, эмулирующую для пользователя среду нейрокомпьютера на обычном компьютере. В настоящее время на рынке программного обеспечения имеется множество самых разнообразных программ для моделирования нейронных сетей. Можно выделить несколько основных функций, которые реализованы во всех этих программах.

1.1. Формирование нейронной сети

Для решения разных практических задач требуются различные модели нейронных сетей. Модель нейронной сети определяется моделями нейронов и структурой связей сети.

Для построения нейронной сети, ориентированной на решение конкретной задачи, используются процедуры формирования нейронных сетей, которые обеспечивают ввод указанных характеристик моделей нейронов и структур нейронных сетей.

Каждая группа моделей нейронных сетей может быть использована для решения лишь некоторого ограниченного класса практических задач. Так, многослойные и полносвязные нейронные сети с сигмоидальными передаточными функциями используются для распознавания образов и адаптивного управления; нейронные сети с локальными связями — для обработки изображений и некоторых других частных задач. Для решения задач линейной алгебры используются многослойные сети с особыми передаточными функциями.

Лишь для небольшого числа моделей нейронных сетей существует строгое математическое обоснование возможности их применения для решения конкретных практических задач. В наибольшей степени теоретически проработаны двухслойные нейронные сети с сигмоидальными переда-

точными функциями. На основе теоремы Колмогорова-Арнольда доказано, что такие сети могут реализовывать любые отображения входного сигнала в выходной. К построению многопараметрических отображений сводится большинство задач распознавания, управления, идентификации.

1.2. Обучение нейронной сети

В большинстве нейропакетов предлагаются стандартные процедуры обучения нейронных сетей, ориентированные на конкретные нейропарадигмы.

Как правило, в нейропакетах реализуется возможность задания различных типов данных и различных размерностей входных и выходных сигналов в зависимости от решаемой задачи. В качестве входных данных в обучающей выборке могут использоваться растровые изображения, таблицы чисел, распределения. Типы входных данных — бинарные (0 и 1), биполярные (-1 и +1) числа, целые или действительные числа из некоторого диапазона. Выходные сигналы сети — векторы целых или действительных чисел.

Для решения практических задач часто требуются обучающие выборки большого объема. Поэтому в ряде нейропакетов предусмотрены средства, облегчающие процесс формирования и использования обучающих примеров. Однако в настоящее время отсутствует универсальная методика построения обучающих выборок, и набор обучающих примеров, как правило, формируется индивидуально для каждой решаемой задачи.

В качестве функции ошибки, численно определяющей сходство всех текущих выходных сигналов сети и соответствующих требуемых выходных сигналов обучающей выборки, в большинстве случаев используется среднеквадратичное отклонение. Однако в ряде нейроимитаторов существует возможность либо выбора, либо задания своей функции ошибки.

Реализуемые в нейропакетах алгоритмы обучения нейронных сетей можно разделить на три группы: градиентные, стохастические, генетические. Градиентные алгоритмы (первого и второго порядков) основаны на вычислении частных производных функции ошибки по параметрам сети. В стохастических алгоритмах поиск минимума функции ошибки ведется случайным образом. Генетические алгоритмы комбинируют свойства стохастических и градиентных алгоритмов: на основе аналога генетического наследования реализуют перебор вариантов, а на основе аналога естественного отбора — градиентный спуск.

1.3. Тестирование обученной нейронной сети

Для проверки правильности обучения построенной нейронной сети в нейроимитаторах предусмотрены специальные средства ее тестирования. В сеть подается некоторый сигнал, который, как правило, не совпадает ни с одним из входных сигналов примеров обучающей выборки. Далее анализируется получившийся выходной сигнал сети.

Тестирование обученной сети может проводиться либо на одиночных входных сигналах, либо на тестовой выборке, которая имеет структуру, аналогичную обучающей выборке, и также состоит из пар (<вход>, <требуемый выход>). Обычно обучающая и тестовая выборки не пересекаются. Тестовая выборка строится индивидуально для каждой решаемой задачи.

2. СРАВНИТЕЛЬНЫЙ АНАЛИЗ НЕЙРОСЕТЕВЫХ ПАКЕТОВ

2.1. NeuroSolutions

Универсальный нейропакет NeuroSolutions фирмы NeuroDimension Inc предназначен для моделирования широкого круга искусственных нейронных сетей. По оценкам специалистов он является лучшим нейросетевым пакетом. Основное достоинство описываемого нейропакета состоит в его гибкости: помимо традиционно используемых нейросетевых парадигм (типа полностью связанных многослойных нейронных сетей или самоорганизующихся полей Кохонена) нейропакет включает в себя мощнейший редактор визуального проектирования нейронной сети, позволяющий создавать практически любые собственные нейронные структуры.

Нейропакет NeuroSolutions снабжен мощными и хорошо продуманными средствами визуализации: отображать в каком угодно виде и визуальным образом контролировать можно все, начиная от структуры нейронной сети и кончая процессом и результатом обучения. Наличие мощных средств визуализации выводит данный нейропакет на уровень САД-систем (систем автоматизированного проектирования), т.е. NeuroSolutions можно считать системой проектирования и моделирования нейронной сети. Помимо грамотно организованных средств взаимодействия с операционной системой (поддерживается OLE2) нейропакет снабжен также генератором исходного кода и средствами, позволяющими использовать внешние модули при проектировании и обучении нейронной сети.

Нейропакет NeuroSolutions имеет также достаточно мощные средства для организации обучающих выборок. Встроенные конверторы данных поддерживают графические изображения в формате BMP, обычные текстовые файлы с числовыми или символьными данными, а также функции непрерывного аргумента (например, времени), заданные в аналитическом виде или в виде выборки значений. Нейропакет позволяет использовать любые внешние конверторы данных, определяемые пользователем.

Для тех, кто использует только традиционные нейронные парадигмы и не пользуется средствами собственного проектирования нейронной сети, нейропакет NeuroSolutions содержит генератор стандартных архитектур (Neural Wizard). С помощью этого генератора можно быстро задать архитектуру нейронной сети, подобрать обучающую выборку, критерии и методы обучения. Поддерживаются наиболее известные нейросетевые парадигмы: многослойные сети, сети Кохонена, самоорганизующиеся структуры и др.

К недостаткам нейропакета NeuroSolutions можно отнести то, что в пакет включено недостаточное количество критериев обучения, если же требуется применять нестандартные критерии обучения, пользователь сталкивается с тем, что ему необходимо создавать внешние модули с последующим подключением к пакету.

В пакете реализовано мало методов обучения: алгоритм обратного распространения, сети Кохонена, самоорганизующиеся структуры. Это наиболее популярные методы, но их явно недостаточно. Для использования других методов обучения пользователю потребуется создать их самому.

В итоге можно сделать вывод, что нейропакет NeuroSolutions имеет множество плюсов и некоторые недостатки и обладает гибкостью и универсальностью.

2.2. NeuralWorks Professional

Нейропакет NeuralWorks Professional так же, как и пакет NeuroSolutions, является мощным средством для моделирования искусственных нейронных сетей. В отличие от пакета NeuroSolutions в NeuralWorks Professional основной упор сделан на применение стандартных нейронных парадигм и алгоритмов обучения, и в этом данный пакет превосходит все остальные. В нем реализованы 28 стандартных нейронных парадигм, используемых при решении прикладных задач. Нейропакет содержит также большое число алгоритмов обучения нейронной сети. Дополнительно поставляемый модуль UDND (User Define Neural Dynamics) позволяет пользователю создавать свои собственные нейронные структуры и работать с ними средствами нейропакета.

Так же, как и NeuroSolutions, NeuralWorks Professional имеет грамотно организованную систему визуализации данных. Можно просмотреть структуру нейронной сети, изменение весовых коэффициентов в процессе обучения, изменение ошибки обучения, а также корреляцию весов нейронной сети при обучении. Последнее является уникальной возможностью, предоставляемой только пакетом NeuralWorks Professional, и весьма полезной при анализе поведения нейронной сети при обучении и работе.

Так же, как и NeuroSolutions, NeuralWorks Professional представляет собой открытую систему, в которую можно интегрировать внешние программные модули, написанные пользователями. Пакет имеет встроенный генератор кода, поддерживающий компилятор Microsoft Visual C++.

Способ представления информации несколько отличается от NeuroSolutions, хотя больших качественных отличий между ними нет. Следует обратить внимание на окно корреляции весовых коэффициентов в левом верхнем углу экрана.

К недостаткам пакета можно отнести то, что он не позволяет пользователям создавать свои нейросети, алгоритмы обучения и даже изменять критерии обучения. Все эти возможности можно реализовать в отдельном модуле UDND. Вторым недостатком является отсутствие мощной системы предварительной обработки данных.

2.3. Process Advisor

Нейропакет Process Advisor специально создавался для решения задач управления динамическими процессами (в частности, технологическими). Однако разработчикам удалось создать программу, которая вполне может считаться нейропакетом.

В данном нейропакете реализована только многослойная нейронная сеть прямого распространения, обучаемая с помощью модифицированного алгоритма обратного распространения ошибки (back-propagation). Поскольку изначально предполагалось, что нейропакет предназначен для решения задач управления, т.е. динамических задач, то в него были введены возможности работы с динамическими процессами.

В частности, в Process Advisor возможна работа с входными сигналами как с функциями времени, а не дискретным набором точек. Такую возможность, помимо Process Advisor, предоставляет только нейропакет NeuroSolutions. Кроме того, нейропакет Process Advisor позволяет осуществлять управление внешними аппаратными контроллерами, подключаемыми к компьютеру. В основном только эти две особенности делают нейропакет Process Advisor примечательным.

Недостатки данного пакета налицо. Нет возможности изменять архитектуру сети, обучающие алгоритмы, критерии обучения. В инструментарии пакета есть только многослойная нейронная сеть, и модифицированный алгоритм обратного распространения ошибки, у которого можно изменять очень ограниченное число параметров. Так же практически отсутствует необходимая система предварительной обработки данных.

2.4. NeuroShell 2

Нейропакет NeuroShell 2 сильно проигрывает по сравнению с NeuroSolutions и NeuralWorks и может считаться универсальным с некоторой натяжкой.

Кроме недостаточно продуманного интерфейса, нейропакет NeuroShell 2 имеет и недостаточно продуманную систему визуализации данных: контролировать можно многие параметры, но в разных режимах работы нейропакета. Из-за отсутствия единого интегрального контроля данных в процессе обучения или работы нейронной сети часто приходится переключаться из одного режима в другой, что отнимает много времени и весьма неудобно в использовании.

К особенностям нейропакета следует отнести жестко реализованную последовательность действий при работе с нейронной сетью. Так, невозможно определить структуру нейронной сети до того, как заданы входные данные. С одной стороны, это очень удобно, особенно для начинающих пользователей, поскольку сразу становится ясно, что и в какой последовательности следует делать. С другой стороны, более опытного пользователя такая жесткая зафиксированная последовательность действий утомляет. Для того, чтобы внести в нейронную сеть небольшое изменение, приходится выполнять всю цепочку действий. Таким образом, можно сказать, что пакет NeuroShell 2 удобен для начинающих пользователей.

В NeuroShell 2 реализована достаточно мощная система обмена данными с другими приложениями. Данный нейропакет позволяет читать данные, представленные в текстовом бинарном виде, а также в наиболее популярных финансовых форматах Mala Stock и Dow Jones. Нейропакет имеет также генератор исходного кода на языках Visual C и Visual Basic

Подводя итог, следует отметить главные недостатки. В пакете нет возможности создания собственной архитектуры нейронной сети, нельзя использовать специфические алгоритмы обучения и изменять критерии обучения нейронной сети.

2.5. Особенности среды Significo

Среда Significo является попыткой создания универсального и гибкого инструментария для работы с нейронными сетями (НС) и различными алгоритмами обработки данных. Ключевыми особенностями среды являются следующие аспекты: среда позволяет создавать системы нейронных сетей, поддерживать 3 уровня пользователей, предусмотреть как использование различных НС и алгоритмов их обучения, так и различных специализированных алгоритмов обработки данных. Рассмотрим более детально эти три основных аспекта.

Возможность организации в среде системы нейронных сетей, содержащей любой элемент среды, очень важна. Практика применения нейронных сетей показывает, что для решения довольно сложных задач требуется применение некой совокупности нейронных сетей в связке с различными специфичными алгоритмами обработки данных. Это позволяет получать не только лучшие результаты, но и верифицировать их. Возможно структурировать решаемую задачу, в результате чего полученное решение можно использовать в будущем, изменяя только часть решения. Также данная методика позволяет интерпретировать результат в логически понятной форме. Затраты ресурсов как временных, так и вычислительных, значительно снижаются, так как приходится обучать несколько нейросетей небольших размеров вместо одной большой сети. Для решения каждой части задачи могут применяться наиболее подходящие элементы среды.

Поддержка трёх уровней пользователей позволяет организовать оптимальный процесс работы исследовательской группы или другого научного коллектива. При данной постановке процесса каждая целевая группа пользователей выполняет свои специфические функции. Предложены три группы с условными названиями: программисты, аналитики, специалисты. Функции трёх целевых групп наиболее удобно рассмотреть на предполагаемом процессе разработки системы нейронных сетей.

- Специалистами составляется техническое задание, включающее общее описание решаемой ими задачи, модель исследуемого объекта, описание входных данных, описание требуемого результата.
- Аналитик или группа аналитиков, изучив техническое задание, составляет спецификацию на создаваемую систему, которая должна включать выбор и обоснование применяемых для решения задачи методов и алгоритмов с учётом таких аспектов, как наличие применяемых методов в среде и др., требуемые затраты ресурсов, спецификации на отсутствующие в среде элементы (для программистов).

стов). Далее спецификация передаётся группе специалистов, и, если у них не возникает никаких замечаний, аналитики передают всю требуемую документацию программистам. Получив недостающие элементы среды, они составляют из элементов систему, настраивают её, тестируют и составляют документацию для пользователей. После чего система нейронных сетей готова к использованию специалистами.

- Программисты, изучив полученную документацию, составляют программную спецификацию на требующиеся аналитикам элементы среды. Далее реализуют элементы, интегрируют их в среду, тестируют и составляют документацию по реализованным элементам.

Как видно из краткого описания технологического процесса, данная поддержка направлена на структуризацию задач, выполняемых в процессе разработки той или иной системы нейронных сетей. Предполагается, что разные этапы различной направленности выполняют участники 3-х целевых групп. Это позволяет выполнять работу в кратчайшие сроки, не отвлекая специалистов от своих специфичных задач. Круг пользователей среды при таком подходе может быть неограниченным.

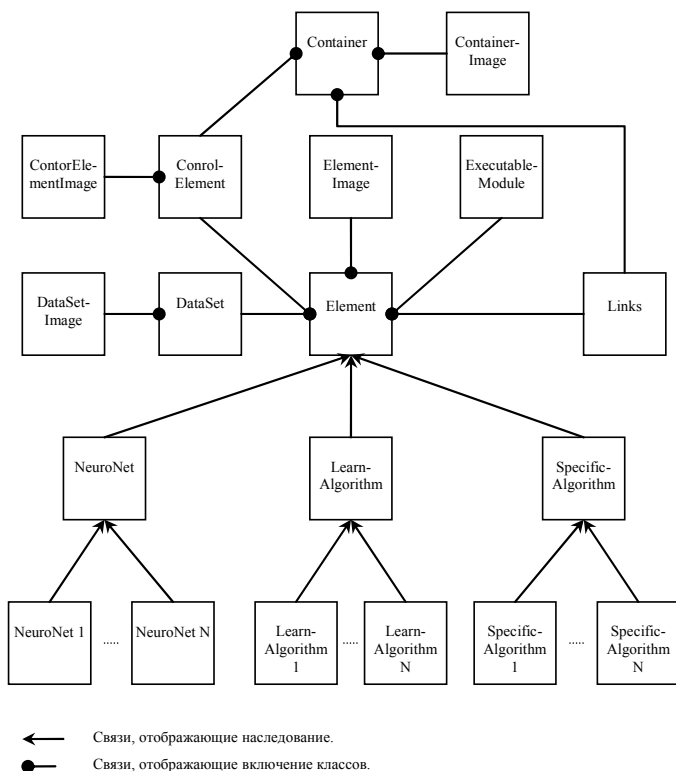
Использование различных алгоритмов обработки данных может значительно расширить область применения среды. Представляется возможным организация мощной системы предварительной обработки данных. Это наиболее важный этап, так как, в основном, любая задача, решаемая с помощью нейросетевых технологий, решается именно на этом этапе. Ведь задача нейросетевого моделирования — найти статистически достоверные зависимости между входными и выходными переменными. Единственным источником информации для статистического моделирования являются примеры из обучающей выборки. С другой стороны, в большинстве случаев при решении комплексных задач данные стекаются со множества нейронных сетей, а далее с помощью детерминированных методов интерпретируются в окончательный результат. Таким образом, возможность использования и дополнения среды различными алгоритмами обработки данных трудно переоценить.

Подводя итоги, хотелось бы отметить, что при использовании среды представляется возможным не только накапливать реализованные в среде методы и алгоритмы, но и осуществлять накопление наработок, решений по различным задачам. Это позволит применять полученные результаты в дальнейшем и распространять их.

3. КРАТКОЕ ОПИСАНИЕ СРЕДЫ

Среду Significo, в соответствии со сравнительным анализом относительно других приложений нейросетевого моделирования, можно классифицировать как универсальный нейросетевой пакет, но с учётом всех особенностей Significo наиболее точным определением является термин «среда нейросетевого моделирования». Для более детального описания среды будут рассмотрены следующие вопросы: базовая архитектура среды, схема взаимодействий среды, подробное описание функционалов для каждой из трёх пользовательских групп.

3.1. Базовая архитектура среды



Функциональное назначение представленных в схеме классов дано в таблице.

Наименование класса	Назначение
ControlElement	Реализует логическую модель групп параметров для управления элементом среды или контейнером. Содержит в себе название параметра, его описание, характеристики.
ControlElementImage	Реализует визуализацию ControlElement. Позволяет пользователю производить манипуляции над ControlElement.
DataSet	Реализует унифицированный набор данных среды, позволяет производить простейшие манипуляции с данными.
DataSetImage	Реализует образ набора данных и позволяет пользователю производить манипуляции с объектом DataSet.
Links	Класс, с помощью которого реализуется взаимодействие между элементами среды и контейнерами.
ExecutableModule	Класс, реализующий логику работы элемента.
Element	Класс, реализующий элемент среды, включает функциональность всех вышеперечисленных классов. Логически его можно представить как совокупность обвязки среды и логическое ядро, которое взаимодействует со средой через интерфейсную оболочку.
ElementImage	Реализует образ элемента среды, включает функции манипуляции элементом, заложенные во всей интерфейсной оболочке элемента среды.
Container	Класс, предназначенный для объединения элементов создаваемой пользователем системы в группы. Данная функция позволяет упрощать графическое представление сложных систем. Вторая функция контейнера — это группировать параметры управления содержащихся в системе элементов.
ContainerImage	Реализует образ контейнера и позволяет им манипулировать в рамках его интерфейса.
NeuroNet	Реализует на базе класса Element класс нейронной сети со всей специфической функциональностью.

LearnAlgorithm	Реализует на базе класса Element класс алгоритма обучения нейронной сети.
SpecificAlgorithm	Реализует на базе класса Element класс алгоритма обработки данных. Это может быть как алгоритм предварительной обработки данных, так и какой-либо специфический алгоритм.
NeuroNet1..NeuroNetN, LearnAlgorithm1..LearnAlgorithmN, SpecificAlgorithm1..SpecificAlgorithmN	Данные классы наследуются от 3-х вышеприведённых классов. Реализуют конкретные нейросетевые архитектуры, алгоритмы обучения и алгоритмы

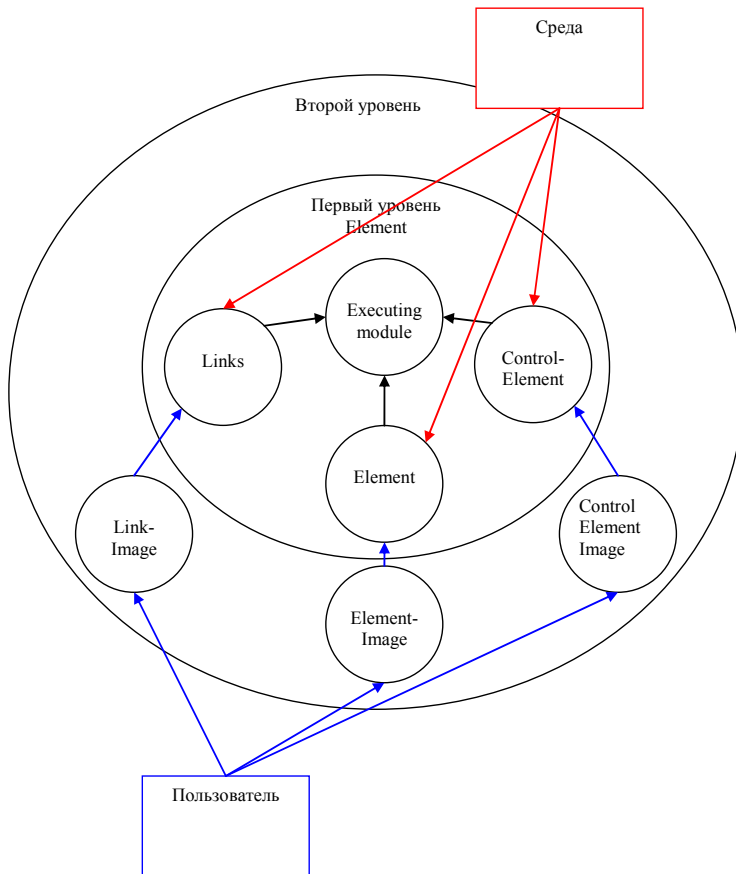
3.2. Схема взаимодействий среды

Центральным элементом среды является класс Element. У данного класса есть две основные области взаимодействия: область взаимодействия с пользователем и область взаимодействия с другими элементами среды.

Область взаимодействия с пользователем организована на классах ControlElementImage, DataSetImage, ContainerImage, ElementImage. Взаимодействие пользователя с элементом производится по цепочке: событие образа, интерфейс класса образа, и замыкает цепочку интерфейс класса Element.

Область взаимодействия между элементами среды осуществляется через одну интерфейсную оболочку.

Таким образом, у элемента среды имеются две интерфейсных оболочки: одна из них участвует при любом взаимодействии, вторая — только лишь при взаимодействии с пользователем.



3.3. Описание пользовательских групп

В среде предусмотрено три группы пользователей. Они поддерживаются путём сокрытия тех частей интерфейса, которые не участвуют в технологических процессах той или иной группы. Если всё же пользователю из какой-либо группы требуется использовать какую-либо функцию, которая не отображается в интерфейсе, то он может легко получить доступ к ней. Разделение интерфейса на 3 группы пользователей обусловлено стремлением к максимальному удобству использования среды. Весь процесс работы со средой также разбит на три уровня: уровень пользователя (специалиста), уровень аналитика и уровень программиста. Рассмотрим более подробно процесс разработки с помощью среды и наиболее точно обозначим функции, выполняемые той или иной группой пользователей.

Группа аналитиков — это основная группа, участвующая в разработке нейросетевых систем среды. Главным направлением данной группы является создание систем нейронных сетей, разработка интерфейса для специалистов, написание документации по проекту. Таким образом, аналитик должен не только ориентироваться в нейросетевых технологиях, но и знать принципы составления интерфейсных XML файлов.

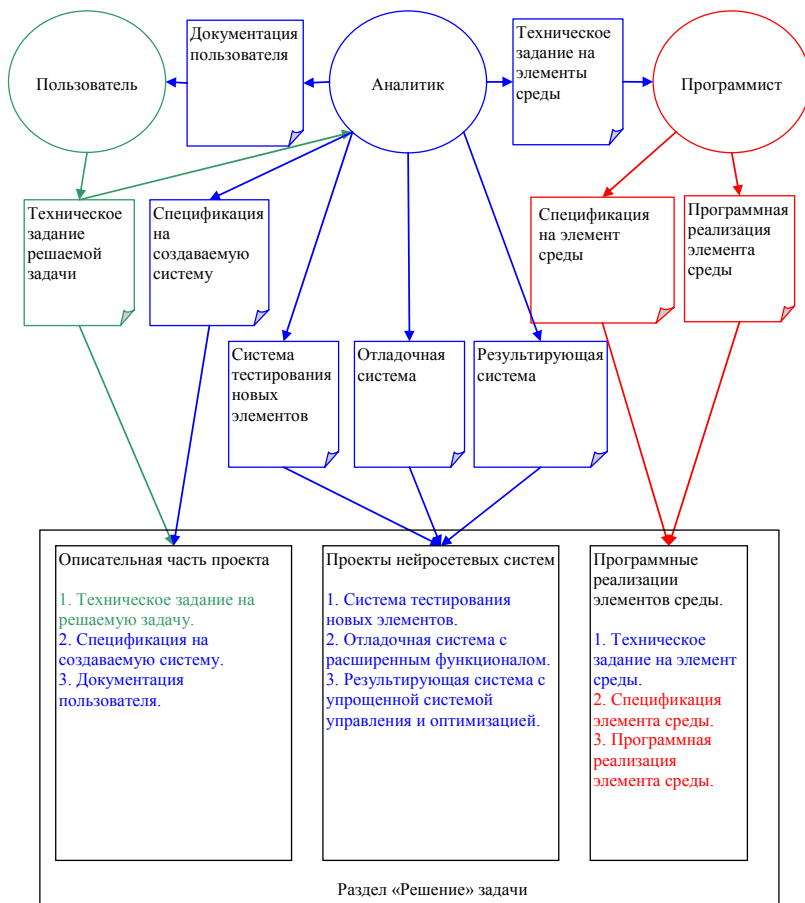
При работе со средой группа программистов не является обязательной, программисты требуются лишь в том случае, если инструментария среды недостаточно для решения поставленной задачи. В противном случае обязанностью программиста будет программная реализация требуемого элемента среды на любом из языков семейства .NET, а также составление интерфейсных XML-файлов.

Группа специалистов представляет собой основную группу пользователей. Для неё использование полученного аналитиками решения должно быть предельно простым. Предполагается, что эта группа пользователей наиболее узкоспециализированная и имеет лишь поверхностные представления о работе среды и нейросетевых технологиях.

Процесс разработки в среде состоит в общем случае из трёх частей. В среде вводятся понятия «Решение» и «Проект». «Решение» должно содержать в себе документацию по решаемой задаче: техническое задание, спецификацию проекта, описание системы, полученной в результате, пользовательскую документацию. Также раздел «Решение» должен включать в себя проекты, созданные в рамках решаемой задачи. Например, при решении задачи потребовалось реализовывать отсутствующие в среде элементы. В этом случае в раздел «Решение» должны быть добавлены проекты тести-

рования добавленных элементов, отладочные системы, используемые аналитиками при решении задачи, и проект с оптимизированной системой для решения поставленной задачи и созданными для пользователей удобными и упрощенными интерфейсами системы.

Весь технологический процесс представлен на следующей схеме.



СПИСОК ЛИТЕРАТУРЫ

1. Горбань А.Н., Дунин-Барковский В.Л., Кирдин А.Н. и др. Нейроинформатика. — Новосибирск: Наука. Сиб. отд-ние РАН, 1998. — 296 с.
2. Уоссермен Ф. Нейрокомпьютерная техника, теория и практика. — Norwood, 1992. — 118 с.
3. Новиков И.С. Обзор нейросетевого программного обеспечения. — 2001. — <<http://vlasov.iu4.bmstu.ru/>>
4. Нейронные сети. — 2004. — <<http://ole-u.cefb.ru/>>
5. Загоруйко Н.Г. Прикладные методы анализа данных и знаний. — Новосибирск: Изд-во Ин-та математики, 1999. — 270 с.
6. Дюк В.А. Data Mining — интеллектуальный анализ данных. — 2003