

*В. Д. Семейкин, А. В. Скупченко*

## **МОДЕЛИРОВАНИЕ ИСКУССТВЕННЫХ НЕЙРОННЫХ СЕТЕЙ В СРЕДЕ MATLAB**

### **Введение**

Теория и аппарат искусственных нейронных сетей (ИНС) и нейросетевых моделей (НСМ) являются активно развивающимся направлением науки и техники. Основные перспективы их использования связаны с решением сложных практических задач. В телекоммуникационных системах они применяются при решении следующих важных задач [1, 2]: управление коммутацией, адаптивная маршрутизация, управление трафиком в телекоммуникационных сетях, планирование сотовых сетей подвижной радиосвязи, оптимальное распределение каналов в сотовых радиосетях.

Решение любой задачи с использованием ИНС и НСМ включает в себя следующие этапы:

- разработка нейросетевой модели;
- формирование входного и желаемого выходного сигналов НСМ;
- формирование сигнала ошибок и функционала оптимизации;
- формирование структуры НСМ, адекватной решаемой задаче;
- разработка алгоритма настройки НСМ, эквивалентного процессу решения задачи в нейросетевом логическом базисе;
- решение задачи с использованием разработанной НСМ.

### **Реализация ИНС в математическом пакете MATLAB**

Искусственные нейронные сети широко используются для решения разнообразных задач. Среди развивающихся областей применения ИНС – обработка аналоговых и цифровых сигналов, синтез и идентификация телекоммуникационных систем. Основы теории и технологии применения ИНС широко представлены в пакете MATLAB. В этой связи особо следует отметить последнюю версию пакета – MATLAB 6.0, где впервые представлен графический интерфейс пользователя GUI (Graphical User Interface) для ИНС – NNTool.

В данной статье дано описание NNTool и показана техника его применения при решении ряда задач синтеза цепей и сетей связи [3].

Применение в телекоммуникационных системах выполняется с помощью либо ИНС без памяти, либо ИНС с памятью. И в том и другом случаях ключевым элементом является ИНС без памяти, подобная роль которой определяется тем обстоятельством, что при использовании нейронов с определёнными функциями активации (передаточными характеристиками) ИНС является универсальным аппроксиматором. Последнее означает, что в заданном диапазоне изменения входных переменных ИНС может с заданной точностью воспроизводить и моделировать произвольную непрерывную функцию этих переменных. Ниже обсуждаются вопросы, относящиеся к так называемым ИНС прямого распространения, т. е. без обратных связей. Замечательным свойством таких ИНС является их устойчивость.

После того как структура ИНС выбрана, должны быть установлены её параметры. Выбор структуры ИНС и типов нейронов – самостоятельный и весьма непростой вопрос, который здесь мы обсуждать не будем. Что же касается значений параметров, то, как правило, они определяются в процессе решения некоторой оптимизационной задачи. Эта процедура в теории ИНС называется обучением.

Графический интерфейс пользователя NNTool позволяет выбирать структуры ИНС из обширного перечня и предоставляет множество алгоритмов обучения для каждого типа сети.

Нами рассматриваются следующие вопросы, относящиеся к работе с NNTool:

- назначение графических управляющих элементов;
- подготовка данных;
- создание нейронной сети;
- обучение сети;
- прогон сети.

### Пример создания нейронной сети

Пусть требуется создать нейронную сеть, выполняющую логическую функцию «И».

Выберем сеть, состоящую из одного персептрона с двумя входами. В процессе обучения сети на её входы подаются входные данные и производится сопоставление значения, полученного на выходе, с целевым (желаемым). На основании результата сравнения (отклонения полученного значения от желаемого) вычисляются величины изменения весов и смещения, уменьшающие это отклонение.

Итак, перед созданием сети необходимо заготовить набор обучающих и целевых данных. Составим таблицу истинности для логической функции «И», где P1 и P2 – входы, а A – желаемый выход (табл.).

Таблица истинности логической функции «И»

P1	P2	A
0	0	0
0	1	0
1	0	0
1	1	1

Данные могут быть представлены любым понятным MATLAB выражением. Например, предыдущее определение вектора целей можно эквивалентно заменить строкой вида `bitand([0 0 1 1], [0 1 0 1])`.

Теперь следует приступить к созданию нейронной сети (рис. 1).

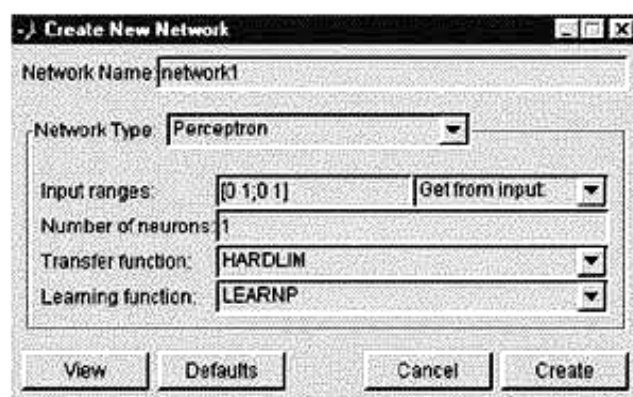


Рис. 1. Окно «Создание сети»

Поля несут следующие смысловые нагрузки:

- имя сети (Network Name) – это имя объекта создаваемой сети;
- тип сети (Network Type) – определяет тип сети и в контексте выбранного типа представляет для ввода различные параметры в части окна, расположенной ниже этого пункта. Таким образом, для разных типов сетей окно изменяет своё содержание;
- входные диапазоны (Input ranges) – матрица с числом строк, равным числу входов сети. Каждая строка представляет собой вектор с двумя элементами: первый – минимальное значение сигнала, которое будет подано на соответствующий вход сети при обучении, второй – максимальное. Для упрощения ввода этих значений предусмотрен выпадающий список «Получить из входа» (Get from input), позволяющий автоматически сформировать необходимые данные, указав имя входной переменной;
- количество нейронов (Number of neurons) – число нейронов в слое;
- передаточная функция (Transfer function) – в этом пункте выбирается передаточная функция (функция активации) нейронов;

– функция обучения (Learning function) – функция, отвечающая за обновление весов и смещений сети в процессе обучения.

Можно посмотреть архитектуру создаваемой сети, т. е. мы имеем возможность удостовериться, все ли действия были произведены верно. На рис. 2 изображена персептронная сеть с выходным блоком, реализующим передаточную функцию с жёстким ограничением. Количество нейронов в слое равно одному, что символически отображается размерностью вектора-столбца на выходе слоя и указывается числом непосредственно под блоком передаточной функции. Рассматриваемая сеть имеет два входа, т. к. размерность входного вектора-столбца равна двум.

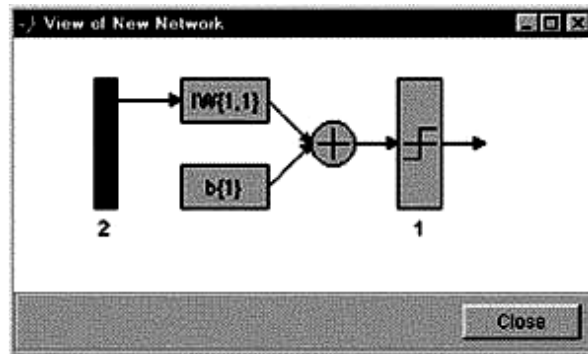


Рис. 2. Предварительный просмотр создаваемой сети

Итак, структура сети соответствует заданию.

В результате проделанных операций в разделе «Сети» (Networks) главного окна NNTool появится объект с именем network1.

Наша цель – построить нейронную сеть, которая выполняет функцию логического «И». Очевидно, нельзя рассчитывать на то, что сразу после этапа создания сети последняя будет обеспечивать правильный результат (правильное соотношение «вход/выход»). Для достижения цели сеть необходимо должным образом обучить, т. е. подобрать подходящие значения параметров. В MATLAB реализовано большинство известных алгоритмов обучения нейронных сетей, среди которых представлено два для персептронных сетей рассматриваемого вида. Вернёмся в главное окно NNTool. На данном этапе интерес представляет нижняя панель «Только сети» (Networks only). Нажатие любой из клавиш на этой панели вызовет окно, на множестве вкладок которого представлены параметры сети, необходимые для её обучения и прогона, а также отражающие текущее состояние сети.

Завершить процесс обучения можно, руководствуясь разными критериями. Возможны ситуации, когда предпочтительно остановить обучение, полагая достаточным некоторый интервал времени. С другой стороны, объективным критерием является уровень ошибки.

На вкладке «Параметры обучения» (Training parameters) для нашей сети (рис. 3) можно установить следующие поля:

- количество эпох (epochs) – определяет число эпох (интервал времени), по прошествии которых обучение будет прекращено (эпохой называют однократное представление всех обучающих входных данных на входы сети);
- достижение цели или попадание (goal) – здесь задаётся абсолютная величина функции ошибки, при которой цель будет считаться достигнутой;
- период обновления (show) – период обновления графика кривой обучения, выраженный числом эпох;
- время обучения (time) – по истечении указанного здесь временного интервала, выраженного в секундах, обучение прекращается.

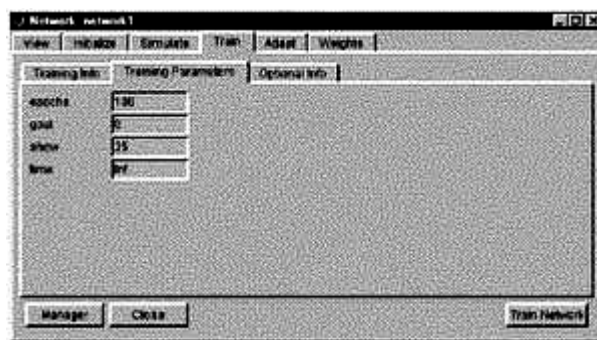


Рис. 3. Вкладка параметров обучения

Принимая во внимание тот факт, что для задач с линейно отделимыми множествами (а наша задача относится к этому классу) всегда существует точное решение, установим порог достижения цели, равный нулю. Значения остальных параметров оставим по умолчанию. Заметим только, что поле времени обучения содержит запись Inf, которая определяет бесконечный интервал времени (от английского Infinite – бесконечный) (рис. 4).

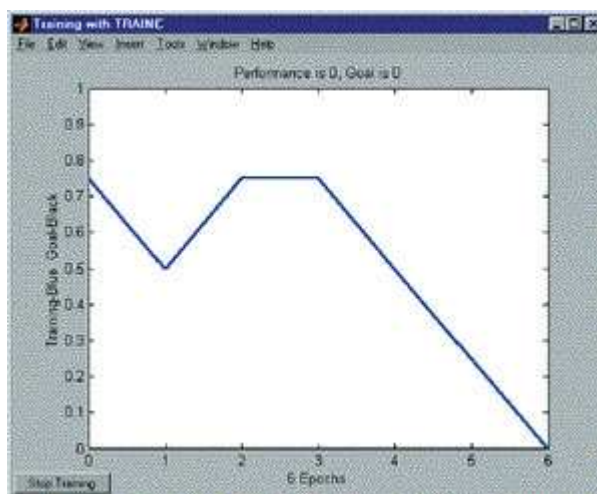


Рис. 4. Кривая обучения

Следует отметить, что для персептронов, имеющих функцию активации с жёстким ограничением, ошибка рассчитывается как разница между целью и полученным выходом.

Итак, алгоритм обучения нашёл точное решение задачи. В методических целях убедимся в правильности решения задачи путём прогона обученной сети. В данной задаче естественно использовать тот же набор данных, что и при обучении data1. Следует заметить, что сеть создаётся инициализированной, т. е. значения весов и смещений задаются определённым образом. Перед каждым следующим опытом обучения обычно начальные условия обновляются.

При выборе нейронной сети для решения конкретной задачи трудно предсказать её порядок. Если выбрать неоправданно большой порядок, сеть может оказаться слишком гибкой и может представить простую зависимость сложным образом. Это явление называется переобучением. В случае сети с недостаточным количеством нейронов, напротив, необходимый уровень ошибки никогда не будет достигнут. Здесь налицо чрезмерное обобщение.

Для предупреждения переобучения применяется следующая техника. В начале работы ошибки сети на обучающем и контрольном множествах будут одинаковыми. По мере того как сеть обучается, ошибка обучения убывает, и, пока обучение уменьшает действительную функцию ошибки, ошибка на контрольном множестве также будет убывать. Если же контрольная ошибка перестала убывать или даже стала расти, это указывает на то, что обучение следует закончить. Остановка на этом этапе называется ранней остановкой (Early stopping).

Таким образом, необходимо провести серию экспериментов с различными сетями, прежде чем будет получена подходящая. При этом, чтобы не быть введённым в заблуждение локальными минимумами функции ошибки, следует несколько раз обучать каждую сеть.

Если в результате последовательных шагов обучения и контроля ошибка остаётся недопустимо большой, целесообразно изменить модель нейронной сети (например, усложнить сеть, увеличив число нейронов, или использовать сеть другого вида). В такой ситуации рекомендуется применять ещё одно множество – тестовое множество наблюдений (Test Data), которое представляет собой независимую выборку из входных данных. Итоговая модель тестируется на этом множестве, что даёт дополнительную возможность убедиться в достоверности полученных результатов. Очевидно, чтобы сыграть свою роль, тестовое множество должно быть использовано только один раз. Если его использовать для корректировки сети, оно фактически превратится в контрольное множество.

Нами рассмотрены простейшие задачи синтеза цепей и сетей, для решения которых применялись наиболее популярные нейронные сети прямого распространения. На самом деле NNTool позволяет решать значительно более широкий круг задач, предоставляя возможность использовать сети разнообразных архитектур, с памятью и без, с обратными связями и без таковых. При этом следует иметь в виду, что успех во многом зависит от понимания поведения конструируемых сетей и их аппроксимационных возможностей.

### **Заключение**

В последние десятилетия в мире отмечается быстрое развитие нейроинформационных технологий. Актуальность исследований в этом направлении подтверждается большим количеством различных применений нейроинформационных систем. Это автоматизация процессов распознавания образов, адаптивное управление, аппроксимация функционалов, прогнозирование, создание экспертных систем и многие другие приложения.

Особый интерес вызывает применение нейроинформационных технологий в инфотелекоммуникационных системах. Высокая эффективность нейроинформационных технологий при решении задач адаптивного управления динамическими системами уже в ближайшее время делает их незаменимыми при создании новых поколений сетей мобильной связи и других беспроводных сетей.

Области применения ИНС и НСМ в телекоммуникационных системах постоянно расширяются. Появление на рынке недорогих многофункциональных и ориентированных на решение конкретных задач нейрочипов и нейрокомпьютеров будет способствовать быстрому переходу к очередному этапу развития телекоммуникационных систем – созданию интеллектуальных телекоммуникационных систем.

### *СПИСОК ЛИТЕРАТУРЫ*

1. *Комашинский В. И., Смирнов Д. А.* Нейронные сети и их применение в системах управления и связи. – М.: Горячая линия – Телеком, 2003. – 94 с.
2. *Семейкин В. Д.* Основные направления применения нейроинформационных технологий в системах связи // Междунар. форум информатизации (МФИ-2005). Междунар. конгресс «Коммуникационные технологии и сети (СТН-2005)». – М.: МТУСИ, 2005. – С. 52–54.
3. *Нейронные сети.* STATISTICA Neural Networks. – М.: Горячая линия – Телеком, 2000. – С. 392.

Статья поступила в редакцию 10.12.2008

## MODELLING OF ARTIFICIAL NEURAL NETWORKS IN MATLAB ENVIRONMENT

*A. V. Skupchenko, V. D. Semeykin*

The object of the research: telecommunication systems, neuronet models and their application to solve telecommunication problems. The purpose of the work: studying and creation of neuronet models to solve telecommunication problems. Neuronet models are studied and developed on the basis of artificial neural networks to solve efficiently the following telecommunication problems: management of switching, management of routing and traffic, distribution of channels in mobile systems of radio communication. Basic technical features: construction of neuronet models on the basis of artificial neural networks and their application to solve telecommunication problems.

**Key words:** modelling, artificial neural networks, telecommunications systems, routing.