

## ПРИМЕНЕНИЕ НЕЙРОННЫХ СЕТЕЙ ДЛЯ АППРОКСИМАЦИИ ДАННЫХ

Рассматривается применение искусственных нейронных сетей для решения задачи аппроксимации данных и сопоставление результатов с традиционным методом построения трендовых моделей.

В последние несколько лет наблюдается взрыв интереса к нейронным сетям, которые успешно применяются в самых различных областях – бизнесе, медицине, технике, геологии, физике. Нейронные сети вошли в практику везде, где нужно решать задачи прогнозирования, классификации или автоматизации. Такой успех определяется несколькими причинами.

1. Богатыми возможностями. Нейронные сети – мощный метод моделирования, позволяющий воспроизводить чрезвычайно сложные зависимости. Нейросети нелинейны по своей природе. Кроме того, нейронные сети справляются с «проклятием размерности», которое не позволяет моделировать линейные зависимости в случае большого числа переменных.

2. Простотой в использовании. Нейронные сети учатся на примерах. Пользователь нейронной сети подбирает представительные данные, а затем запускает алгоритм обучения, который автоматически воспринимает структуру данных. От пользователя, конечно, требуется какой-то набор эвристических знаний о том, как следует отбирать и подготавливать данные, выбирать нужную архитектуру сети и интерпретировать результаты, однако уровень знаний, необходимый для успешного применения нейронных сетей, гораздо скромнее, чем, например, при использовании традиционных методов статистики.

### 1. ОСНОВНЫЕ ПОНЯТИЯ

Нейронная сеть – это система, состоящая из многих простых вычислительных элементов, работающих параллельно, функция которых определяется структурой сети, силой взаимосвязанных связей, а вычисления производятся в самих элементах или узлах.

Нейронная сеть – это набор нейронов, определенным образом связанных между собой. Работу искусственного нейрона можно описать следующим образом [1] (рис. 1).

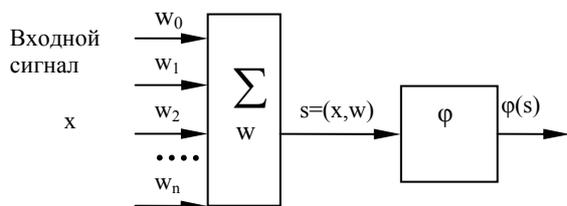


Рис. 1. Нейрон

Нейрон получает входные сигналы (исходные данные либо выходные сигналы других нейронов нейросети) через несколько входных каналов. Каждый входной сигнал проходит через соединение, имеющее определенную интенсивность (или вес); этот вес соответствует синаптической активности биологического нейрона. С каждым нейроном связано определенное пороговое значение. Вычисляется взвешенная сумма входов, из нее вычитается пороговое значение, и в результате получается величина активации нейрона.

Сигнал активации преобразуется с помощью функции активации (или передаточной функции). В результате получается выходной сигнал нейрона:

$$s = (x, w) = w_0 + \sum_{i=1}^n x_i w_i.$$

Наиболее распространенными функциями активации являются следующие: пороговая, сигнум (или модифициро-

ванная пороговая функция), логистическая, гиперболический тангенс, линейная, линейная ограниченная, радиально-базисная и др. Как правило, передаточные функции всех нейронов в сети фиксированы, а веса являются параметрами сети и могут изменяться. Если сеть предполагается для чего-то использовать, то у нее должны быть входы (принимающие значения интересующих нас переменных из внешнего мира) и выходы (прогнозы или управляющие сигналы). Кроме этого в сети может быть еще много промежуточных (скрытых) нейронов, выполняющих внутренние функции. Входные, скрытые и выходные нейроны должны быть связаны между собой.

Наиболее распространенными являются многослойные сети, в которых нейроны объединены в слои. Слой – это совокупность нейронов, на которые в каждый такт времени параллельно поступает информация от других нейронов сети. После того, как определено число слоев и число элементов в каждом из них, нужно найти значения для весов и порогов сети, которые минимизировали бы ошибку прогноза, выдаваемого сетью. Именно для этого служат алгоритмы обучения. С использованием собранных исторических данных веса и пороговые значения автоматически корректируются с целью минимизации этой ошибки. По сути, этот процесс представляет собой подгонку модели, которая реализуется сетью, к имеющимся обучающим данным. Ошибка для конкретной конфигурации сети определяется путем прогноза через сеть всех имеющихся наблюдений и сравнения реально выдаваемых выходных значений с желаемыми (целевыми) значениями. Все такие разности суммируются в функцию ошибок, значение которой и есть ошибка сети. В качестве функции ошибок чаще всего берется сумма квадратов ошибок:

$$E(w) = \sum_i (d_i - y_i)^2, \quad (1)$$

где  $d$  – желаемый выход сети,  $y$  – реальный выход сети.

Самый известный вариант алгоритма обучения нейронной сети – так называемый алгоритм обратного распространения [2]. Существуют современные алгоритмы второго порядка, такие, как метод сопряженных градиентов и метод Левенберга–Маркара [3], которые на многих задачах работают существенно быстрее (иногда на порядок). Алгоритм обратного распространения в некоторых случаях имеет определенные преимущества.

### 2. ПОСТАНОВКА ЗАДАЧИ

Рассмотрим задачу аппроксимации ряда динамики, т.е. построение функции по конечному набору точек, на реальном примере о количестве проданных авиабилетов за 3 года (данные приведены по месяцам на рис. 2,  $N = 36$  – объем выборки).

Приступая к разработке нейросетевого решения, как правило, сталкиваются с проблемой выбора оптимальной архитектуры нейронной сети. Так как области применения наиболее известных парадигм пересекаются, то для решения конкретной задачи можно использовать различные типы нейронных сетей, и при этом результаты могут оказаться

одинаковыми. Будет ли та или иная сеть лучше и практичнее, зависит в большинстве случаев от условий задачи.

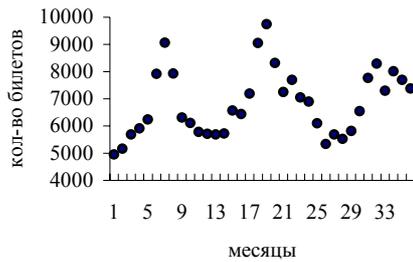


Рис. 2. Периодический ряд динамики продаж авиабилетов

Задачи аппроксимации экспериментальных данных можно решать с помощью искусственных нейронных сетей следующих типов: многослойного персептрона, сетей с радиально-базисными функциями, вероятностных сетей, обобщенно-регрессионных сетей [1,4].

Решим поставленную задачу с помощью нейронных сетей типа радиальная базисная (RBF), обобщенно-регрессионная (GRNN) и линейной сети. Моделирование решения задач проводилось с применением пакета прикладных программ Neural Network Toolbox (NNT) системы MATLAB [4].

Искусственные нейросети выходные значения представляют, как правило, в определенном диапазоне или масштабируют. Выполним предобработку данных – нормировку исходных данных  $y_i$  ( $t = \overline{1, N}$ ), для этого воспользуемся стандартной формулой:

$$y'_i = \frac{y_i - My}{\sqrt{Dy}}, \quad (2)$$

где  $y'_i$  – новая переменная;  $My = \frac{1}{n} \sum_i y_i$  – выборочная оценка математического ожидания;  $Dy = \frac{1}{n-1} \sum_i (y_i - My)^2$  – выборочная оценка дисперсии.

Существуют и другие функции масштабирования, из них самая простейшая – минимаксная функция, выполняющая линейное преобразование после определения минимального и максимального значений функции так, чтобы получаемые значения находились в нужном диапазоне. Линейное преобразование  $\tilde{y}_i = \frac{y_i - y_{i,\min}}{y_{i,\max} - y_{i,\min}}$  масштабирует исходную выборку в единичный отрезок [0, 1].

Предобработку полезно провести для того, чтобы повысить скорость и качество обучения.

Задача аппроксимации функции для нейронной сети формируется как задача контролируемого обучения (обучение с учителем). Суть задачи состоит в следующем. Имеются значения функции в отдельных точках (узлах), система базисных функций и векторов регулируемых весовых коэффициентов. Необходимо обучить сеть, т. е. выбрать весовые коэффициенты при базисных функциях так, чтобы их комбинация давала аналогичную зависимость, которая наилучшим образом аппроксимирует множество значений функции отклика.

### 3. ПРИМЕНЕНИЕ

#### РАДИАЛЬНО-БАЗИСНОЙ НЕЙРОННОЙ СЕТИ

Сеть типа радиально-базисной функции (RBF) имеет промежуточный слой из радиальных элементов, каждый из которых воспроизводит гауссову поверхность отклика:

$$\varphi(s) = \exp(-s^2/2\sigma^2), \quad (3)$$

где параметр  $\sigma$  определяет радиус влияния каждой базисной функции и быстроту стремления к нулю при удалении от центра (рис. 3). Поскольку эти функции нелинейны, для моделирования произвольной функции нет необходимости брать более одного промежуточного слоя. Для моделирования любой функции необходимо лишь взять достаточное число радиальных элементов. Остается решить вопрос о том, как следует скомбинировать выходы скрытых радиальных элементов, чтобы получить из них выход сети. Оказывается, что достаточно взять их линейную комбинацию (т.е. взвешенную сумму гауссовых функций). Сеть RBF имеет выходной слой, состоящий из элементов с линейными функциями активации

$$\varphi(s) = ks, \quad (4)$$

где  $k$  – угловой коэффициент наклона прямой.

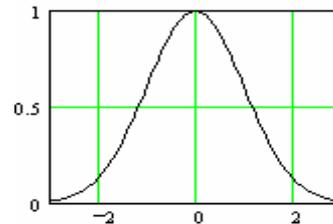


Рис. 3. Радиальная базисная функция

Процесс обучения RBF-сетей включает две стадии: процесс настройки центров базисных функций (3) и обучение нейронов в скрытом слое, поэтому эти сети обучаются достаточно быстро.

Для решения поставленной задачи была создана RBF-сеть (рис. 4) с допустимой среднеквадратической ошибкой  $E(\omega)=0,3$  и параметром влияния, равным 1, значение которого устанавливается тем больше, чем больший диапазон входных значений должен быть принят во внимание. В качестве входных элементов было взято время  $t = \overline{1, N}$ , а выходы – уровни временного ряда  $y'_i$ . Функции пакета NNT, используемые при создании RBF-сети: net = newrb(t, y'; 0,3; 1) – создание радиальной базисной нейронной сети с обучением; un' = sim(net, t) – моделирование сети.

Чтобы полученный выход сети соответствовал реальному (истинному) масштабу данных, было выполнено обратное преобразование:

$$yn_i = un'_i \sqrt{Dy} + My, \quad (5)$$

где  $yn_i$  – преобразованный выход сети;  $un'_i$  – выход сети;  $My$  и  $Dy$  описаны выше.

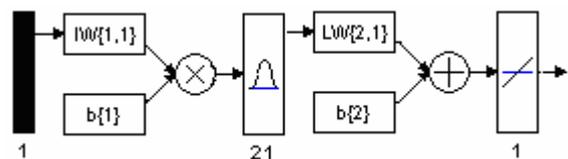


Рис. 4. Нейронная сеть с радиально-базисными элементами

В результате моделирования RBF-сети для реальных данных была получена аппроксимирующая функция (рис. 5). На рис. 6 приведен график соответствующей ошибки (отклонения фактических данных от расчётных)  $\varepsilon_t = y_t - y_{nt}$ .

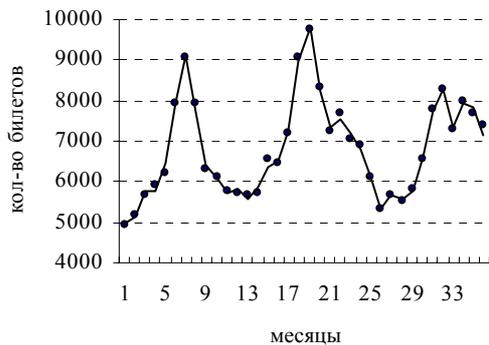


Рис. 5. Аппроксимация данных с помощью радиально-базисной сети



Рис. 6. Ошибка аппроксимации данных RBF-сетью

#### 4. ПРИМЕНЕНИЕ ОБОБЩЕННО-РЕГРЕССИОННОЙ НЕЙРОННОЙ СЕТИ

Обобщенно-регрессионная нейронная сеть (GRNN), как и RBF-сети, имеет радиально-базисный слой с числом нейронов, равных числу элементов или менее обучающего множества, но еще включает линейный слой. GRNN-сеть копирует внутрь себя все обучающие наблюдения и использует их для оценки отклика в произвольной точке. Окончательная выходная оценка сети получается как взвешенное среднее выходов по всем обучающим наблюдениям, где величины весов отражают расстояние от этих наблюдений до той точки, в которой производится оценивание (и, таким образом, более близкие точки вносят больший вклад в оценку).

Первый промежуточный слой сети GRNN состоит из радиальных элементов. Второй промежуточный слой (линейный) содержит элементы, которые помогают оценить взвешенное среднее. Для этого используется специальная процедура. Каждый выход имеет в этом слое свой элемент, формирующий для него взвешенную сумму. Чтобы получить из взвешенной суммы взвешенное среднее, эту сумму нужно поделить на сумму весовых коэффициентов. Последнюю сумму вычисляет специальный элемент второго слоя. После этого в выходном слое производится собственно деление (с помощью специальных элементов «деления»). Таким образом, число элементов во втором промежуточном слое на единицу больше, чем в выходном слое. Как правило,

в задачах регрессии (аппроксимации) требуется оценить одно выходное значение и, следовательно, второй промежуточный слой содержит два элемента.

Процесс обучения GRNN-сети аналогичен обучению RBF-сети. Первоначально настраиваются центры базисных функций (3), затем с фиксированными параметрами RBF-нейронов обучается выходной слой.

Для решения поставленной задачи была смоделирована GRNN-сеть с нулевой ошибкой  $E(\omega) = 0$  и коэффициентом сглаживания, равным 1, наибольшее значение (равное 1) которого соответствует наиболее сглаженному ряду (рис. 7). В качестве входных элементов было взято время  $t = \overline{1, N}$ , а выхода – уровни временного ряда  $y'_t$ . Функции пакета NNT:  $net = \text{newgrnn}(t, y', 1)$  – создание GRNN-сети;  $yn' = \text{sim}(net, t)$  – моделирование сети.

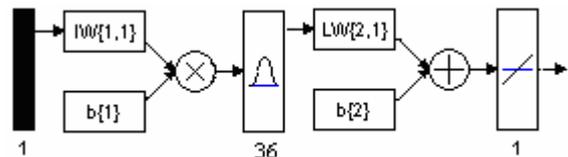


Рис. 7. Обобщенно-регрессионная нейронная сеть

Полученная аппроксимирующая функция исследуемых данных с помощью GRNN-сети представлена на рис. 8, причем предварительно было сделано обратное преобразование (5). График соответствующей ошибки (отклонения фактических данных от расчётных)  $\varepsilon_t = y_t - y_{nt}$  дан на рис. 9.

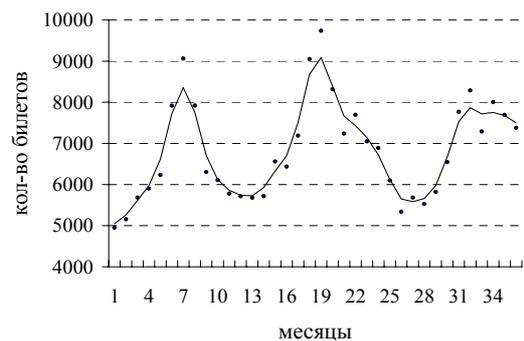


Рис. 8. Аппроксимация данных с помощью обобщенно-регрессионной сети

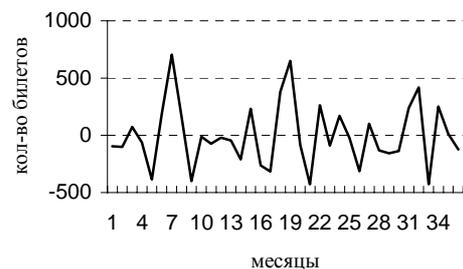


Рис. 9. Ошибка аппроксимации данных GRNN-сетью

#### 5. ПРИМЕНЕНИЕ ЛИНЕЙНОЙ НЕЙРОННОЙ СЕТИ

Линейная сеть – это самая простая нейросеть, которая не содержит промежуточных (скрытых) слоев и в выходном слое содержит только линейные элементы (т.е. элементы с линейной функцией активации (4)). Но для решения поставленной задачи использовалась модифицированная сеть – линейная сеть с задержкой на входе (рис. 10).

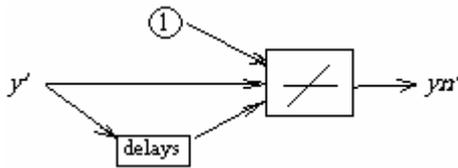


Рис. 10. Линейная нейронная сеть с задержкой на входе

Для рассматриваемого ряда динамики характерны внутригодовые, повторяющиеся устойчиво из месяца в месяц изменения в уровнях, т.е. сезонные колебания. Поэтому предлагается использовать линейную сеть с задержкой на входе шести предыдущих входных элементов. Входными величинами этой сети являются уровни временного ряда  $y'_t$ , получаемые преобразованием (2). Выходной слой сети будет содержать один элемент, соответствующий выровненному значению ряда. Для обучения такой сети требуется большой объём выборки, более 100, поэтому в качестве обучающей выборки использован исходный ряд, повторенный 4 раза, при этом  $N = 144$ .

Фрагмент программы создания и адаптации линейной сети:

```
delays = [1 2 3 4 5 6] – вектор задержки;
net=newlin([-2 2.8], 1, delays, 0.01);
in=con2seq(y'); d=in;
nnet=adapt(net, in, d);
yn'=sim(nnet, d).
```

Процесс адаптации (приспособления) линейной сети к данным приведен на рис. 11.

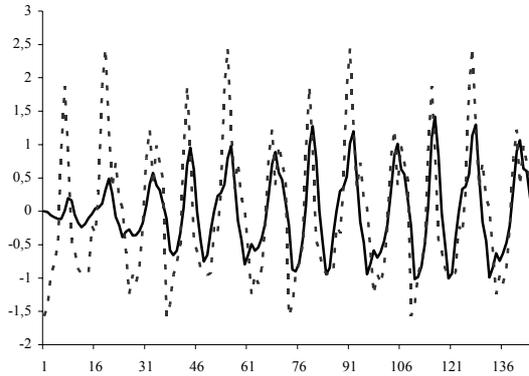


Рис. 11. Адаптация линейной сети к обучающей выборке

После того как сеть «адаптируется», её можно использовать для интерполяции и экстраполяции данных.

Результат аппроксимации данных с применением линейной сети приведен на рис. 12.

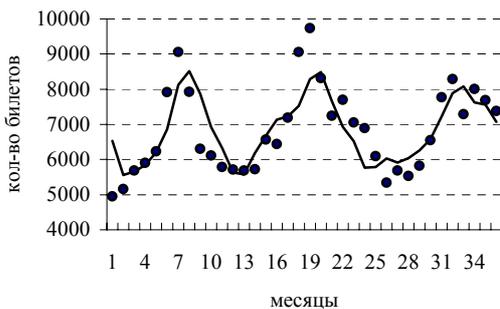


Рис. 12. Аппроксимация данных

Проанализируем полученные результаты сглаживания ряда динамики традиционным методом аналитического выравнивания, т.е. построения трендовой модели. Динамический ряд (рис. 1) является стационарным периодическим рядом, т.е. общей тенденции развития нет, но явно выделяется циклическая составляющая временного ряда и, естественно, случайная компонента. Поэтому для сглаживания данного РД были использованы в качестве модели тренда ряд Фурье и мультипликативная модель [5]. Ряд Фурье или модель тригонометрического тренда имеет вид

$$y_t = a_0 + \sum_{i=1}^n a_i \cos \frac{2\pi i}{N} t + \sum_{i=1}^n b_i \sin \frac{2\pi i}{N} t, \quad (6)$$

где  $a_0, a_i, b_i$  – неизвестные параметры,  $n = 6$  – число гармоник,  $N = 36$  – длина ряда динамики (объём выборки). Для решения задачи идентификации параметров данного уравнения применяли классический метод наименьших квадратов [5]. Графики исходного и выровненного рядов динамики приведены на рис. 13.

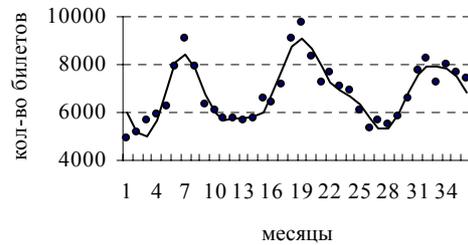


Рис. 13. Сглаживание ряда динамики рядом Фурье

При мультипликативной модели уровень РД можно представить как произведение его составляющих:

$$y_t = \bar{y} \cdot \frac{\bar{y}_s}{\bar{y}} \cdot \frac{y_t}{\bar{y}_s}, \quad t = \overline{1, N}, \quad (7)$$

где отношение  $\bar{y}_s / \bar{y}$  представляет собой коэффициент сезонности ( $K_s$ ), а  $y_t / \bar{y}_s$  – отражает влияние случайного фактора,  $\bar{y}_s$  – средний уровень ряда соответствующего периода внутри года (месяца, квартала) за ряд лет.

Оценочные значения для модели (7) можно представить в виде

$$\hat{y}_t = \bar{y} \cdot (K_s)_t, \quad t = \overline{1, N}. \quad (8)$$

Графики исходного и выровненного рядов динамики приведены на рис. 14.

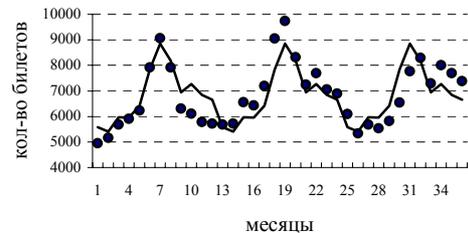


Рис. 14. Сглаживание ряда динамики с применением мультипликативной модели

Качество аппроксимации различными методами оценивалось по коэффициенту детерминации (табл. 1), ко-

торый характеризует так называемую долю «объясненной» дисперсии и определяется как

$$R^2 = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - My)^2},$$

где  $y_i$  – исходные значения,  $\hat{y}_i$  – выровненные значения,  $My$  – выборочное среднее исходного ряда динамики. Качество аппроксимации тем лучше, чем ближе этот коэффициент к 1.

Т а б л и ц а 1

	Ряд Фурье	Мультипликативная модель	GRNN-сеть	RBF-сеть	Линейная сеть
Коэффициент детерминации ( $R^2$ )	0,8747	0,7262	0,9473	0,9918	0,6422

Неудовлетворительное качество аппроксимации с применением линейной сети объясняется её простотой и спецификой исходного ряда динамики, который включает сезонные колебания. При отсутствии ярко выраженных колебаний можно предпочесть использование линейной сети, так как при этом обеспечивается сглаживание данных от случайной составляющей.

По приведенным данным можно сделать вывод о том, что применение нейронных сетей даёт приемлемый (достаточно высокий) уровень аппроксимации исходных наблюдаемых данных, прежде всего это благодаря наличию в RBF и GRNN нейросетях скрытого слоя нейронов с нелинейными радиально-базисными функциями активации, которые позволяют отслеживать малейшие изменения в уровнях исследуемого временного ряда. При использовании RBF-сети получили  $R^2 = 0,9918$ ; это случай когда реальный выход нейросети и желаемый выход (что по смыслу совпадает с оценочными и реальными значениями) практически совпадают. Применяя традиционные методы, практически невозможно достичь такого высокого значения коэффициента детерминации.

Любой ряд динамики содержит случайную составляющую, поэтому наиболее приемлемо использование

метода, который выявляет общую тенденцию развития исследуемого ряда, так как при прогнозировании наиболее важно учесть закономерности изменения уровня ряда, а не случайные явления, влияющие на исследуемый показатель. Поэтому в рассматриваемом примере наиболее предпочтительным вариантом является использование GRNN-сети, так как она обеспечивает высокое качество аппроксимации и показывает тенденции изменения показателя в наиболее продолжительных отрезках времени.

Результаты применения традиционных методов сглаживания ряда динамики по сравнению с использованием нейронных сетей типа RBF и GRNN не намного уступают по качеству. Если нет возможности применять нейросети, то предпочтительнее использовать ряд Фурье для аппроксимации исследуемых данных, а не мультипликативную модель.

Таким образом, для рассматриваемого примера аппроксимации данных по известным объемам продаж авиабилетов применение нейронных сетей обеспечивает высокое качество аппроксимации и может использоваться для анализа и прогнозирования деятельности авиапредприятий.

#### ЛИТЕРАТУРА

1. *Нейронные сети* Statistica neural Networks. М.: Телеком, 2000.
2. *Организация и обучение искусственных нейронных сетей* / Авт.-сост. Л.В. Калацкая, В.А. Новиков, В.С. Садков. Мн.: БГУ, 2003.
3. *Bishop C.* Neural networks for pattern recognition. Oxford: University Press, 1995.
4. *Медведев В. С., Потемкин В. Г.* Нейронные сети. MATLAB 6. М.: Диалог – МИФИ, 2002.
5. *Статистика: Учебник* / Под ред. проф. И.И. Елисеевой. М.: ООО «ВИТРЭМ», 2002.

Статья представлена кафедрой прикладной математики факультета прикладной математики и кибернетики Томского государственного университета, поступила в научную редакцию «Кибернетика» 20 апреля 2004.