

## РАЗРАБОТКА И АПРОБАЦИЯ ЭМУЛЯТОРА НЕЙРОСЕТЕВОГО МОДЕЛИРОВАНИЯ ДЛЯ ЦЕЛЕЙ ПРОГНОЗИРОВАНИЯ ВРЕМЕННЫХ РЯДОВ

Дмитриев Д.В., Ляхманов Д.А., Соколова Э.С.<sup>1</sup>

<sup>1</sup> ФГБОУ ВПО «Нижегородский Государственный Технический Университет Им. Р.Е.Алексеева» Нижний Новгород, Россия (603950, Нижний Новгород, ГСП-41, ул. Минина, д. 24) ), e-mail: [dmitdmit@mail.ru](mailto:dmitdmit@mail.ru)

Представлено решение актуальной задачи автоматизации разработки архитектуры нейронных сетей для целей прогнозирования с последующей апробацией на временных рядах различной природы. Учитывались следующие характеристики нейронных сетей на этапе проектирования и обучения: число слоев сети, число нейронов в каждом слое, тип активационной функции, коэффициент обучаемости, условия окончания обучения. На основе результатов, полученных в ходе нейросетевого моделирования, представлены рекомендации по обработке исходной информации для прогноза. Проанализированы возможности точного прогнозирования будущих результатов поведения финансового рынка, с учетом внутренних, неочевидных факторов и влияния внешней среды. Представлены результаты работы эмулятора по прогнозированию финансовых временных рядов котировок акций на долгосрочном периоде.

Ключевые слова: нейронные сети, прогнозирование, финансовые временные ряды, моделирование.

## DEVELOPMENT AND TESTING OF NEURAL NETWORK MODELING EMULATOR FOR TIME SERIES PREDICTION

Dmitriev D.V., Lahmanov D.A. , Sokolova E.S.<sup>1</sup>

<sup>1</sup> "Nizhny Novgorod State Technical University n.a. R.E. Alekseev" Nizhny Novgorod, Russia (603950, Nizhny Novgorod, street Minin, 24), e-mail: [dmitdmit@mail.ru](mailto:dmitdmit@mail.ru)

We introduced the solution of the actual problem of automating design architecture of neural networks for forecasting, followed by appraisal at the time series of different nature. We took into account the following characteristics of neural networks in the design phase and training: the number of network layers, each layer of the number of neurons, the type of activation function, learning rate, the terms of graduation. Based on the results obtained from neural network modeling, provide recommendations for handling input to the forecast. The possibilities of accurate prediction of future performance of financial markets, including domestic, non-obvious factors and the influence of the external environment. We introduced the results of the emulator on forecasting financial time series of stock price in the long term are at the end.

Key words: neural networks, forecasting, financial time series, modeling.

### Введение

В современном мире часто приходится принимать решения в условиях неопределенности, например, на основе построенного прогноза. Для решения задач прогноза, наряду с классическими методами статистического прогнозирования, моделирования, экспертного поиска решения [2-4], успешно используются нейросетевые технологии [1,5], представляющие перспективное направление развития искусственного интеллекта.

Нейронные модели используются при решении широкого диапазона задач прогнозирования, аппроксимации, распознавания образов и т.д. Решение задач выполняется в два этапа – проектирование сети и ее обучение для конкретных целей.

Для реализации нейронных сетей используются аппаратные средства на основе дорогостоящих нейронных плат, а также программы-эмуляторы с возможностью адаптации к любой предметной области.

### **Цель исследования**

На сегодняшний день существует ряд программных продуктов, предназначенных для моделирования работы нейронных сетей. Это NeuroPro, Neuro Office, Deductor Studio и многие другие. Каждая из этих программ представляет собой достаточно мощное средство проектирования, но в ряде случаев весьма ощутимы их недостатки: ограничение по классам задач, жестко установленные конфигурации сетей, отсутствие возможности изменения различных настроек (активационная функция, алгоритмы обучения и т.д.). Например, ограниченная функциональность (закрытые программные коды) не позволяет эффективно применять существующие средства проектирования для задач прогнозирования, в которых динамически изменяются условия проведения прогноза. Кроме того, не все программы имеют дружелюбный, понятный пользователю интерфейс с визуальным отображением построенной сети. Часто отсутствует возможность легкого импортирования обучающих и тестовых последовательностей. И, наконец, достаточно серьезное ограничение по применению – высокая стоимость этих программ.

Для устранения большей части недостатков, указанных выше, разработан новый адаптивный метод, реализованный в виде программного обеспечения и апробированный при прогнозировании финансовых временных рядов. Для этих целей разработан новый тип данных, описывающий нейронную сеть без привязки к конкретной конфигурации. При этом пользователю обеспечена возможность варьирования настройки и параметров не только самой сети, но и параметров процесса обучения. Все это представлено в виде интуитивно-понятного дружелюбного интерфейса.

### **Разработка архитектуры эмулятора нейросетевого моделирования**

Иерархия классов построена в соответствии с архитектурой нейронной сети, представленной множеством слоев. Каждый слой представляет собой объединение большого числа нейронов, общее количество ограничено поставленной задачей и вычислительными ресурсами. При этом нейроны разных слоев связаны друг с другом определенным образом (нейросеть может быть полносвязная или связи могут быть заданы самим пользователем). В редких случаях возможна связь нейронов одного слоя между собой, что предусмотрено в разработанной модели.

Пользователю разработанного программного продукта предоставлена возможность самостоятельно задать структуру и конфигурацию сети (количество слоев, количество

нейронов в каждом слое, связи между нейронами) под решение своей задачи, что позволяет повысить эффективность и точность работы сети в дальнейшем.

Для организации связей между нейронами чаще всего используют массив весов, но в данном программном продукте для этих целей спроектирован свой собственный класс, позволяющий эффективно работать как со связями, так и с весовыми коэффициентами на этапах проектирования, обучения и эксплуатации сети.

После задания структуры и конфигурации сети проводится процесс ее обучения, представляющий собой поиск оптимальных значений весовых коэффициентов синоптических связей. Разработанный метод адаптируется к решению различных прикладных задач. В этой связи пользователю программы предоставлены различные алгоритмы обучения с настройкой параметров обучения, т.к. от качества обучения напрямую зависит способность сети решать поставленные задачи.

В результате исследований в среде программирования Code Gear Delphi 2009 построен эмулятор нейросетевого моделирования NeuroFriend, разработка которого представлена следующими этапами.

На первом этапе спроектирована иерархия классов, представляющая собой упрощенную модель функционирования нейронной сети.

Общепринято в основу иерархии ставить класс **TNeuron** (Нейрон), что увеличивает сложность модификации разработанных программных средств при адаптации к меняющимся условиям. В предложенном методе в основу иерархии поставлен класс **TLink** (Связь), описывающий связи между нейронами. Данная особенность позволила получить простую, наглядную, вложенную классовую архитектуру. Внутри класса **TNeuron** описаны все методы и поля, необходимы для расчета выходного значения на основе входных данных, значений весовых коэффициентов и типа активационной функции.

Чтобы задать данные о структуре и конфигурации сети, разработан класс **TLayer** (Слой), хранящий массив нейронов соответствующего слоя сети. И, наконец, глобальный класс **TNet** (Нейронная сеть) содержит общую информацию о нейронной сети. Внутри этого класса инкапсулированы все ее параметры: массив слоев, данные об активационной функции, диапазоны входных и выходных значений, характеристики обучения и многое другое, что является общим для всей сети. Кроме того, в классе **TNet** описаны основные инструменты работы с сетью: методы обучения сети, методы расчета выходных сигналов, методы начальной инициализации весовых коэффициентов, методы расчета ошибок и т.д.

В результате проектирования получена иерархия классов, представленная на рис. 1.

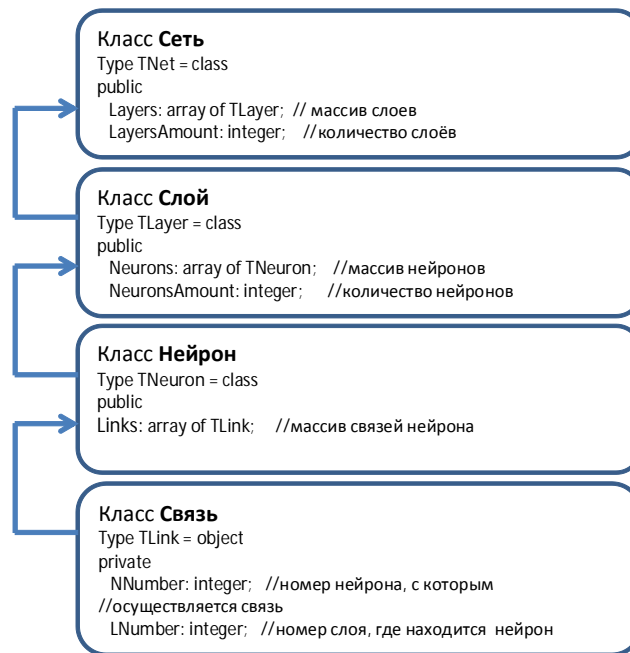


Рис. 1. Иерархия классов для эмулятора нейросетевого моделирования

На втором этапе программно реализованы алгоритмы обучения сети, адаптированные под разработанную иерархию классов: алгоритм обратного распространения ошибки (как наиболее простой и эффективный алгоритм обучения с учителем) и дифференциальный метод Хебба (как пример обучения без учителя). Отметим, что любой другой алгоритм обучения может быть интегрирован в эмулятор в виде отдельной библиотеки.

Если процесс обучения попадает в локальный минимум, когда заданная точность не достигнута, но ошибка не перестает уменьшаться (это касается алгоритма обратного распространения), производится кратковременное увеличение шага обучения, позволяющее выйти из локального минимума. Данная возможность используется автоматически или по запросу, что выгодно отличает NeuroFriend от других программ.

Адаптированный алгоритм обратного распространения включает следующие шаги:

1. На входы сети подается один из тестовых образов. Сигнал распространяется в прямом направлении от входов к выходам. В процессе распространения рассчитывается выходной сигнал каждого нейрона. Расчет состояния нейрона ведется по формуле:

$$s_j^{(n)} = \sum_{i=0}^M y_i^{(n-1)} \cdot w_{ij}^{(n)},$$

где  $M$  – число нейронов в слое  $n-1$ ;  $y_i^{(n-1)} = x_{ij}^{(n)}$  –  $i$ -й вход нейрона  $j$  слоя  $n$ ;

$y_j^{(n)} = f(s_j^{(n)})$ , где  $f()$  – активационная функция;

$y_q^{(0)} = I_q$ , где  $I_q$  –  $q$ -я компонента вектора входного образа.

2. В соответствии с имеющимся образцом рассчитывается ошибка  $\delta^{(N)}$  и производится коррекция весовых коэффициентов  $\Delta w^{(n)}$  выходного слоя по формулам:

$$\delta_j^{(n)} = \left[ \sum_k \delta_k^{(n+1)} \cdot w_{jk}^{(n+1)} \right] \cdot \frac{dy_j}{ds_j},$$

$$\Delta w_{ij}^{(n)} = -\eta \cdot \delta_j^{(n)} \cdot y_i^{(n-1)}$$

3. Внутренние слои последовательно корректируются в направлении от выходов к входам по формуле:

$$w_{ij}^{(n)}(t) = w_{ij}^{(n)}(t-1) + \Delta w_{ij}^{(n)}(t)$$

4. В случае попадания в локальный оптимум производится изменение шага обучения  $\eta$ , которое задается автоматически или пользователем. В качестве  $\eta$  выбирается значение в интервале  $[0,1]$ . После того, как значения весовых коэффициентов стабилизируются, следует кратковременное значительное увеличение  $\eta$ , чтобы начать градиентный поиск глобального оптимума из новой точки. Пользователем задается количество повторов процедуры увеличения шага обучения, приводящей в одно и то же состояние нейронной сети, при достижении которого принимается решение о нахождении глобального, а не локального оптимума.

5. При достижении заданной точности или в случае превышения максимального количества прогонов – окончание обучения. В противном случае возврат к первому шагу.

Адаптированный алгоритм Хебба реализован также в виде отдельного метода и выглядит следующим образом:

1. На входы сети подается один из тестовых образов. Сигнал распространяется в прямом направлении от входов к выходам. В процессе распространения рассчитывается выходной сигнал каждого нейрона.

2. Производится корректировка весовых коэффициентов на основе текущих состояний нейронов и полученных выходных значений по формуле:

$$w_{ij}(t) = w_{ij}(t-1) + \alpha \cdot y_i^{(n-1)} \cdot y_j^{(n)},$$

где  $y_i^{(n-1)}$  – выходное значение нейрона  $i$ -го слоя ( $n-1$ ),  $y_j^{(n)}$  – выходное значение нейрона  $j$ -го слоя  $n$ ;  $w_{ij}(t)$  и  $w_{ij}(t-1)$  – весовой коэффициент синапса, соединяющего эти нейроны, на итерациях  $t$  и  $t-1$  соответственно;  $\alpha$  – коэффициент скорости обучения. Под  $n$  подразумевается произвольный слой сети.

3. Если выходные значения аксонов выходного слоя изменились на величину меньшую, чем требуемая точность, или в случае превышения максимального количества прогонов – окончание обучения. В противном случае возврат к первому шагу.

Большое внимание в процессе разработки эмулятора нейросетевого моделирования NeuroFriend уделено реализации дружественного интерфейса программы. Большая часть основных действий с программой осуществляется с помощью манипулятора «мышь». При этом многие функции могут быть вызваны также с помощью нажатия «горячих» клавиш. Наличие подсказок и информационной панели внизу рабочего окна программы облегчает работу начинающему пользователю. Пункты основного меню четко сгруппированы в соответствии с функциональностью по разделам. Это позволяет легко ориентироваться в программе и быстро находить нужные методы. По такому же принципу сгруппированы кнопки на панели быстрого доступа.

Важным моментом является контроль вызова функций в зависимости от контекста. Иными словами, те функции, которые нельзя вызвать по смыслу на данном этапе работы с сетью, будут неактивны и недоступны для вызова, что отобразится пользователю как неактивная кнопка или неактивный пункт меню.

Этапы создания и обучения четко разделены:

1. Задание конфигурации сети (количество входов, выходов, слоев и нейронов в слоях).
2. Выбор активационной функции и соответствующих параметров (при этом вид активационной функции выводится на экран в виде графика).
3. Графическое представление нейронной сети. Пользователь может задать начальные значения весовых коэффициентов синоптических связей.

Для наглядного отображения сети и простого управления разработан ряд алгоритмов, упрощающих работу пользователя.

Этап обучения сети:

1. Выбор алгоритма обучения сети и задание соответствующих параметров обучения.
2. Задание обучающего и тестового множества (реализована возможность загрузки данных из текстовых файлов).
3. Указание дополнительных параметров обучения (точность обучения и максимальное количество итераций).

Задав все параметры, по нажатию соответствующей клавиши, запускается процесс обучения, в ходе которого во вспомогательном окне отображаются основные данные о текущем состоянии нейросети.

По окончании процесса строится диаграмма работы сети на тестовом множестве.

Также, используя инструмент визуального отображения структуры сети, пользователь может ознакомиться со значениями весов, полученными в процессе обучения сети.

Спроектированную сеть можно сохранить в специально разработанном для этих целей формате в файле с расширением *.net*, с текстовым содержимым которого можно ознакомиться через стандартный блокнот.

Для решения задач прогнозирования в эмуляторе NeuroFriend разработан метод скользящего окна. Для этого загружаются исходные данные из файла, затем указываются глубина погружения (количество отсчётов, которые используются в качестве входных данных) и горизонт прогнозирования. На выходе нейросети получаем прогнозируемое значение.

### Результаты исследования

Практическую апробацию эмулятор NeuroFriend прошел при решении актуальной задачи прогнозирования временных финансовых рядов – сложной динамической системы, как правило, не поддающейся формализации. Чтобы корректно спрогнозировать цену, следует спрогнозировать поведение участников рынка. Это возможно, если выявить скрытые закономерности коллективной психологии путем анализа данных по прошлым торгам. Следует отметить, что принимаемые решения субъективны и не всегда могут быть объяснены логически.

Результат работы обученной нейросети на тестовом множестве представлен на рис. 2.

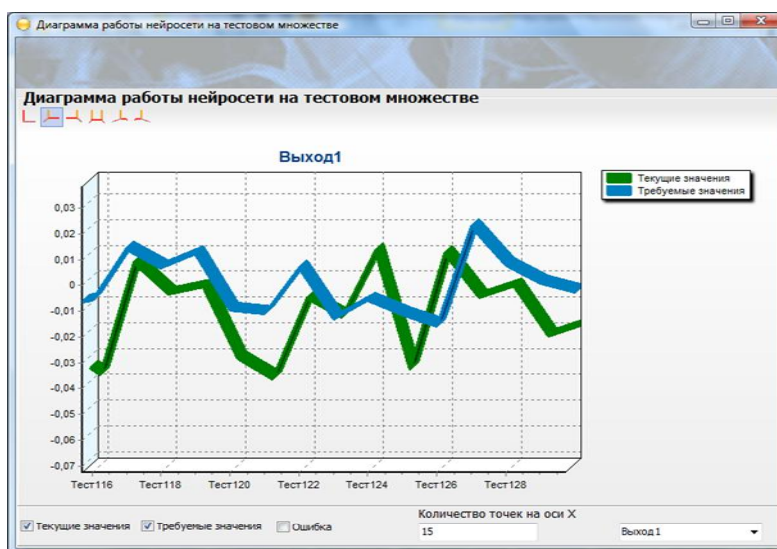


Рис. 2. Нейросеть с 80-ю входами, 1-м выходом, 3-мя скрытыми слоями: 20x8x2

В большинстве из проделанных опытов, были получены положительные результаты. По результатам полученных прогнозов хорошо отслеживается динамика процесса и моменты разворотов трендов, что является хорошим результатом, т.к. целью прогнозирования временных финансовых рядов является максимизация прибыли. Поэтому, в прогнозировании временных финансовых рядов, важна не близость прогноза к истинному значению, а

одинаковая направленность прогноза и истинного значения. Количество верно угаданных тенденций рынка составляло 70...85 %.

## **Выводы**

При правильном обучении нейросети все ее основные ошибки связаны с недостатком входной информации и глобальным изменением состояния объекта прогнозирования (влияние новостей и внешних событий).

Проведенные эксперименты показали, что применение эмулятора нейросетевого моделирования NeuroFriend для целей прогнозирования временных финансовых рядов вполне оправдано и способствует получению прибыли во время работы на финансовых рынках.

## **Список литературы**

1. Бухнин А.В., Бажанов Ю.С. Оптимизация баз знаний экспертных систем с применением нейронных нечетких сетей // Нейрокомпьютеры: разработка, применение. – М.: Радиотехника, 2007. №11
2. Дмитриев Д.В., Капранов С.Н, Соколова Э.С. Адаптация генетических алгоритмов к решению задач назначения точек контроля в объектах с большим числом состояний // Нейрокомпьютеры: разработка, применение. – М.: Радиотехника, 2007. №11.
3. Капранов С.Н., Соколова Э.С. Разработка моделей и методов обеспечения контролепригодности систем с помощью эволюционного моделирования. Труды IV Международной конференции «Идентификация систем и задачи управления» SICPRO '05 Москва 25 января – 28 января 2005 г. С.637-658.
4. Соколова Э.С., Ляхманов Д.А. Метод прогнозирования на основе анализа динамики спектральных составляющих // Информационно-измерительные и управляющие системы. – М.: Радиотехника. №3, 2011.
5. Ширяев В.И. Нейросетевые методы в анализе финансовых рынков // учеб. пособие. М.: КомКнига, 2007.

## **Рецензенты:**

Федосенко Юрий Семенович, д.т.н., профессор кафедры Информатики, систем управления и телекоммуникаций, Волжская государственная академия водного транспорта, г.Нижний Новгород.

Хранилов Валерий Павлович, д.т.н., профессор кафедры Компьютерных технологий в проектировании и производстве, Нижегородский Государственный Технический Университет им. Р.Е. Алексеева, г.Нижний Новгород.

Антонов Александр Владимирович, д.т.н., профессор, декан факультета «Кибернетики» Обнинского института атомной энергетики Национального исследовательского ядерного университета МИФИ Министерства Образования и Науки Российской Федерации. г.Обнинск.