М.А. Моцаков

О ПРОБЛЕМЕ ПРОГНОЗА ВРЕМЕННЫХ РЯДОВ ПРИ ПОМОЩИ ИСКУССТВЕННЫХ НЕЙРОННЫХ СЕТЕЙ

M.A. Motsakov

ON THE PROBLEM OF TIME SERIES FORECASTING BASED ON ARTIFICIAL NEURAL NETWORKS

В данной работе раскрыта проблема построения прогностических моделей временных рядов на основе искусственных нейронных сетей. Предлагается алгоритм создания простой нейросети и ее обучения, а также алгоритм прогноза временного ряда. Приведены результаты экспериментов с моделью.

The problem of construction of forecast models on the basis of artificial neural networks is considered. An algorithm is offered for creation of simple neural networks and its training, as well as an algorithm of time series forecasting. The results of experiments with the models are shown.

Одним из популярных подходов к решению задачи прогноза является представление поведения некоторого процесса в виде временного ряда. Тогда задача прогноза решается, как задача экстраполяции временного ряда, на основе модели, построенной по результатам его анализа. Методы, применяемые для анализа и прогноза временного ряда, определяются с одной стороны, целями анализа и заблаговременностью прогноза, а с другой стороны, вероятностной природой формирования его значений. Укажем самые распространенные методы анализа и прогноза временных рядов [Поляк, 1989]:

Методы анализа:

1. Спектральный анализ: позволяет находить периодические составляющие временного ряда.
2. Корреляционный анализ: позволяет находить существенные периодические зависимости и соответствующие им задержки как внутри одного ряда (автокорреляция), так и между несколькими рядами (кросскорреляция).

Методы прогноза:

1. Модели авторегрессии и скользящего среднего.
2. Многоканальные модели авторегрессии и скользящего среднего: модели применяются в тех случаях, когда имеется несколько коррелированных между собой временных рядов.
3. Прогноз при помощи экспоненциально-взвешенного скользящего среднего. Простейшая модель прогнозирования временного ряда.
4. Модели на основе искусственных нейронных сетей. Аналог моделей авторегрессии, возможно более эффективный.

В настоящее время все большее распространение получают алгоритмы анализа и прогноза на основе искусственных нейронных сетей. Любая нейронная сеть используется в качестве самостоятельной системы представления знаний, которая в практических приложениях выступает, как правило, в качестве одного из компонентов системы управления, либо модуля принятия решений, передающих результирующий сигнал на другие элементы, не связанные непосредственно с искусственной нейронной сетью.

Выполняемые сетью функции можно распределить на несколько основных групп: аппроксимация и интерполяция; распознавание и классификация образов; сжатие данных; прогнозирование; идентификация; управление; ассоциация. В каждом из названных приложений нейронная сеть играет роль универсального «аппроксиматора» функции от нескольких переменных, реализуя нелинейную функцию $Y = f(x)$, где x – это входной вектор, а Y-реализация векторной функции нескольких переменных [Осовский, 2004].

Искусственный нейрон имитирует в первом приближении свойства биологического нейрона, и выступает в качестве элементарного процессора способного к простейшей обработке информации. На вход искусственного нейрона поступает некоторое множество сигналов. Каждый сигнал умножается на соответствующий вес, и все произведения суммируются, определяя уровень активации нейрона. На рис. 1 представлена модель, реализующая эту идею. Хотя сетевые парадигмы весьма разнообразны, в основе почти всех лежит эта конфигурация. На рис. 1 множество входных сигналов, обозначенных x_1, x_2,…, x_n, поступает на искусственный нейрон. Эти входные сигналы в совокупности, обозначим вектором – X. Каждый сигнал умножается на соответствующий вес w_1, w_2,…, w_n, и поступает на суммирующий блок, обозначенный – $\sum$. Множество весов в совокупности образует вектор – W. Суммирующий блок складывает взвешенные входные сигналы, создавая выходной сигнал, который мы будем называть – NET. В векторных обозначениях это может быть компактно записано следующим образом: NET = XW. Блок F представляет собой сжимающую функцию (например, гиперболический тангенс) [Уоссермен, 1992].

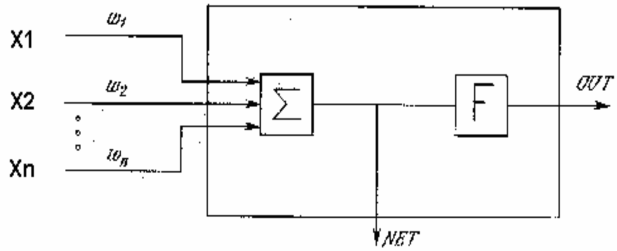

Рис. 1. Искусственный нейрон

Хотя один нейрон и способен выполнять простейшие процедуры преобразования информации, сила нейронных вычислений проистекает от соединения многих нейронов в единую сеть. Простейшая сеть состоит из группы нейронов, образующих слой, рис. 2. Отметим, что вершины-круги слева служат лишь для распределения входных сигналов. Они не выполняют каких-либо вычислений, и поэтому не считаются слоем. По этой причине они обозначены кругами, чтобы отличать их от вычисляющих нейронов, обозначенных квадратами.

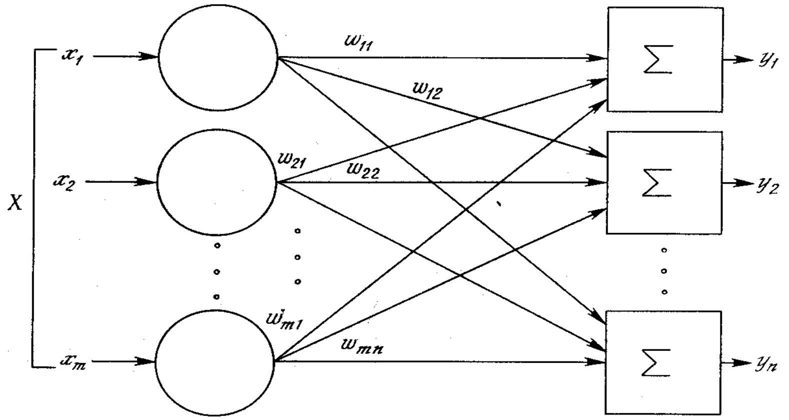

Рис. 2. Однослойная нейронная сеть

Каждый элемент из множества входов – X, отдельным весом соединен с каждым искусственным нейроном. А каждый нейрон выдает взвешенную сумму входов в сеть. В искусственных и биологических сетях многие соединения могут отсутствовать, все соединения показаны в целях общности. Могут иметь место также соединения между выходами и входами элементов в слое.

Более крупные и сложные многослойные сети могут образовываться каскадами слоев. Такие нейронные сети обладают, как правило, большими вычислительными возможностями. Послойная организация нейронов копирует слоистые структуры определенных отделов мозга. Выход одного слоя является входом для последующего слоя.

Целью обучения сети является такая подстройка ее весов, чтобы приложение некоторого множества входных сигналов приводило к требуемому множеству сигналов на выходе. Для краткости эти множества входов и выходов называются векторами.

На рис. 3 изображена многослойная сеть на основе простейших сигмоидальных нейронов, которая может быть обучена при помощи процедуры обратного распространения. Первый слой нейронов, как уже отмечалось, служит лишь в качестве распределительных точек. А каждый нейрон последующих слоев выдает сигналы NET и OUT.

При обучении предполагается, что для каждого входного вектора существует парный ему выходной (целевой) вектор. Вместе они называются обучающей парой. Применительно к задаче прогноза временного ряда, входным вектором будет являться часть ряда предшествующая настоящему моменту времени, за исключением некоторого количества последних значений ряда используемых в качестве целевого вектора сети.

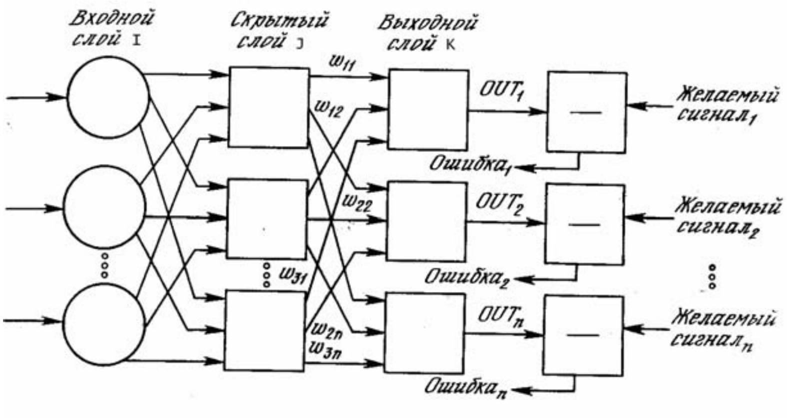

Рис. 3. Двухслойная сеть обратного распространения

Перед началом обучения всем весам должны быть присвоены небольшие начальные значения, выбранные случайным образом. Это гарантирует, что в сети не произойдет насыщения большими значениями весов. Например, если всем весам придать одинаковые начальные значения, а для требуемого функционирования нужны неравные значения, то сеть не сможет обучиться.

Обучение сети обратного распространения требует выполнения следующих операций:

1. Выбрать обучающую пару из обучающего множества (временного ряда).
2. Подать входной вектор на вход сети.
3. Вычислить выход сети.
4. Вычислить разность между выходом сети и требуемым выходом (целевым вектором обучающей пары).
5. Подкорректировать веса сети так, чтобы минимизировать ошибку.
6. Повторять шаги с 3 по 5 до тех пор, пока ошибка не достигнет приемлемого уровня.

Операции, выполняемые шагами 1 и 3, сходны с теми, которые выполняются при функционировании уже обученной сети, т. е. подается входной вектор и вычисляется выходной вектор. Вычисления выполняются послойно (см. рис. 3). Сначала вычисляются выходы нейронов слоя – J, затем они используются в качестве входов слоя – K, вычисляются выходы нейронов слоя K, которые и образуют выходной вектор сети.

На шаге 4 каждый из выходов сети, которые на рис. 3 обозначены OUT, вычитается из соответствующей компоненты целевого вектора, чтобы получить ошибку. Эта ошибка используется на шаге 5 для коррекции весов сети, причем знак и величина изменений весов определяются алгоритмом обучения (см. ниже).

После достаточного числа повторений этих пяти шагов разность между действительными выходами и целевыми выходами должна уменьшиться до приемлемой величины, при этом говорят, что сеть обучилась.

Шаги 1 и 3 можно рассматривать как «проход вперед», так как сигнал распространяется по сети от входа к выходу. Шаги 4, 5 составляют «обратный проход», здесь вычисляемый сигнал ошибки распространяется обратно по сети и используется для подстройки весов.

Проход вперед.

Шаги 1 и 3 могут быть выражены в векторной форме следующим образом: подается входной вектор X и на выходе получается вектор Y. векторная пара вход-цель X и Y берется из обучающего множества.

Как мы видели, вычисления в многослойных сетях выполняются слой за слоем, начиная с ближайшего к входу слоя. Величина NET каждого нейрона первого слоя вычисляется как взвешенная сумма входов нейрона. Затем активационная функция F «сжимает» NET и выдает величину OUT для каждого нейрона в этом слое. Когда выходной вектор слоя получен, он является входным вектором для следующего слоя.

Процесс повторяется слой за слоем, пока не будет получен заключительный выходной вектор сети. Этот процесс может быть выражен в сжатой форме. Например, вес от нейрона 8 в слое I к нейрону 5 слоя J обозначается $w_{8,5}$. Тогда NET-вектор слоя J может быть выражен, не как сумма произведений $\sum x_i w_i$, где i – номер нейрона в слое I (i = 1, 2, 3, ... N, N – количество нейронов в слое I), а как скалярное произведение векторов X и W. В векторном обозначении $\mathrm{NET}_{ns} = X_{ns}W_{ns}$, где ns – индекс слоя ($ns = I, J, K$). Покомпонентным применением функции F к NET-вектору получается выходной вектор – OUT. Таким образом, для данного слоя вычислительный процесс описывается следующим выражением

$$OUT_{ns} = F_{ns}(OUT_{ns\text{-}1} \cdot W_{ns}),\ ns = I, J, K. \tag{1}$$

Обратный проход.

Подстройка весов выходного слоя: Так как для каждого нейрона выходного слоя задано целевое значение, то подстройка весов легко осуществляется с использованием величины δ, равной разности между целевым выходом – TG, и реальным выходом выходного слоя OUT

$$\delta_{i,k} = (TG - \mathrm{OUT}_{i,k}),\ i = 1, 2, 3, \ldots N. \tag{2}$$

Внутренние слои называют «скрытыми слоями», для их выходов не имеется целевых значений для сравнения. Поэтому обучение усложняется. Вычитая значение выхода нейрона слоя K из целевого значения TG, получаем сигнал ошибки. Он умножается на производную сжимающей функции $\mathrm{t}anh(OUT)' = 1 - OUT^2$, вычисленную для этого нейрона слоя K давая, таким образом, величину δ.

$$\delta_{i,k} = (1 - OUT_{i,k}^{\ 2})\,(TG - OUT_{i,k}). \tag{3}$$

Затем δ умножается на величину OUT нейрона слоя – J. Это произведение в свою очередь умножается на коэффициент скорости обучения η (обычно от 0,01 до 1,0), и результат прибавляется к весу нейрона слоя – K. Такая же процедура выполняется для каждого веса от нейрона скрытого слоя к нейрону в выходном слое. Следующие уравнения иллюстрируют это вычисление:

$$\delta w_{i,k} = \eta\, \delta_{i,k}\, OUT_{i,j}; \tag{4}$$

$$w_{i,k}(t+1) = w_{i,k}(t) + \delta w_{i,k}, \tag{5}$$

где $w_{i,k}(t)$ – величина веса нейрона выходного слоя до коррекции (на итерации t); $w_{i,k}\,(t+1)$ – величина веса после коррекции (на итерации $t+1$); $\delta_{i,k}$ – величина ошибки для нейрона – i, в выходном слое – K; $OUT_{i,j}$ – значение на выходе нейрона – i в скрытом слое – J.

Подстройка весов скрытого слоя: Рассмотрим один нейрон в скрытом слое, предшествующем выходному слою. При проходе вперед этот нейрон передает свой выходной сигнал нейронам в выходном слое. Во время обучения сеть функционирует в обратном порядке, пропуская величину δ от выходного слоя назад к скрытому слою. Каждый из весов выходного слоя умножается на величину δ нейрона в выходном слое. Величина δ, необходимая для нейрона скрытого слоя, получается суммированием всех таких произведений и умножением на производную сжимающей функции:

$$\delta_{i,j} = (1 - OUT_{i,j}{}^{2})\left[\sum_{\mathrm{i}} \delta_{i,k} w_{i,k}\right]. \tag{6}$$

Когда значение δ получено, веса первого скрытого уровня, могут быть подкорректированы с помощью уравнений (4) и (5), где индексы модифицируются в соответствии со слоем.

Для каждого нейрона в скрытом слое должно быть вычислено δ и подстроены все веса связанные с этим слоем. Этот процесс повторяется слой за слоем по направлению к входу, пока все веса не будут подкорректированы.

В процессе обучения сети значения весов могут в результате коррекции стать очень большими. Это может привести к тому, что все или большинство нейронов будут функционировать при очень больших значениях OUT, в области, где производная сжимающей функции очень мала. Так как посылаемая обратно в процессе обучения ошибка пропорциональна этой производной, то процесс обучения может практически замереть. В теоретическом отношении эта проблема плохо изучена. Обычно этого избегают уменьшением размера коэффициента - η, но это увеличивает время обучения. В данной работе использован метод сжатия весов при помощи функции вида: $w_1 = 0{,}5\left[1 + \exp(-w_0)\right]$, где w_0 – вес до сжатия, w_1 – после сжатия. Сжатие веса производится по превышении им значения величины порога сжатия (в данной работе порог сжатия выбран равным – 1,0)

Важным этапом в решении задачи прогноза временного ряда при помощи нейросети является формирование обучающего вектора. От состава, полноты, качества входного вектора в значительной мере зависят время обучения сети и достоверность получаемых моделей поведения физического процесса, представленного временным рядом. Для большинства нейронных сетей характерно наличие интервала допустимых значений входных сигналов, в пределах которого сигналы различимы. Функция активации устанавливает допустимые границы значений исходных данных. Отображение в этот диапазон в основном осуществляется с помощью простейшего преобразования – нормализации. В данной работе применен метод нормализации входного вектора вида (7).

$$X_i^{norm} = \frac{X_i}{\sqrt{\sum_{i=1}^{N} X_i^2}}. \qquad (7)$$

Далее изложены основные принципы построения прогностического алгоритма временного ряда на основе нейросети.

На этапе обучения сети происходит формирование матрицы весовых коэффициентов. То есть происходит самоорганизация нейромодели, способной прогнозировать временной ряд на один шаг вперед. Обучение происходит до тех пор, пока ошибка обучения (EL), равная разности между значением выхода сети – OUT и целевым значением временного ряда – A[learn], не будет достаточно малой, или перестанет изменять свою величину. Либо пока *EF* – функция ошибки прогноза по тестовому значению(-ям) – A[fore], не достигнет локального минимума (рис. 4).

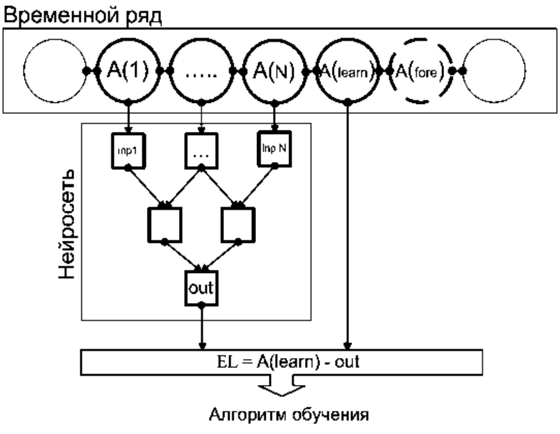

Рис. 4. Обучение сети

По окончании обучения, входы нейросети «смещаются» относительно временного ряда на одно значение вперед. То есть на вход обученной сети подается «неизвестный» ей сигнал, который, проходя через нее, формирует выходную величину – OUT. Таким образом, мы получаем прогностическое значение временного ряда. Ошибка прогноза – EF, определяется как квадрат разности величин A[fore] и OUT (рис. 5).

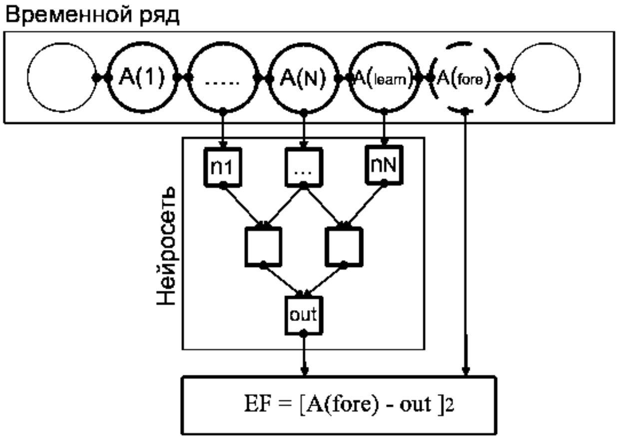

Рис. 5. Прогноз значения временного ряда

Смещая нейросеть пошагово вдоль временного ряда и, подавая на входы сети, спрогнозированные, ею же значения, мы получаем механизм прогноза на необходимое количество шагов во времени (рис. 6).

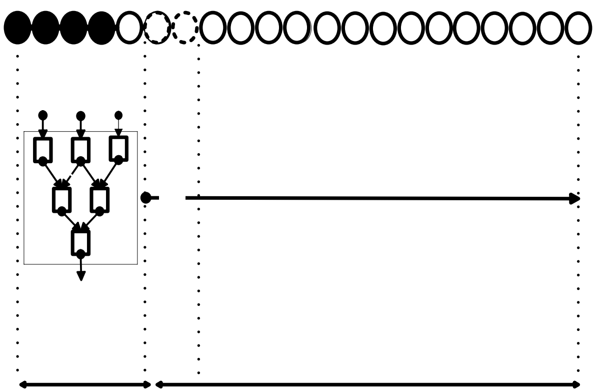

Рис. 6. Схема прогноза временного ряда при помощи нейросети

В данной работе рассматривается построение модели временного ряда на основе искусственной нейронной сети. Создание подобных алгоритмов является необходимым, в связи с бурным развитием систем автоматического измерения параметров окружающей среды. Нейромодель может быть использована в процессе оценки изменения состояния окружающей среды и принятия оперативных решений в качестве, вспомогательного альтернативного традиционным методам, инструмента.

Задача, поставленная автором в данной работе, может быть сформулирована следующим образом:

Некая система автоматической регистрации производит измерение величин ограниченного числа параметров с относительно малой дискретностью во времени, в фиксированной точке пространства. Прогностическая группа за неимением или недостатком дополнительной информации, вынуждена использовать в качестве опорных или вспомогательных данных для принятия оперативных решений, результаты измерений вышеописанной системы.

Усложним задачу прогноза, сократив количество измеряемых параметров до одного. В качестве простейших методов получения прогностической информации в данной ситуации, можно рассматривать любые экстраполяционные алгоритмы, начиная с построения линейного тренда и заканчивая моделями на основе авторегрессии. Последние отличаются громоздкостью и неудобны при использовании в оперативном режиме. Более гибкие и более эффективные нейромодели, непрерывно усваивая поток информации, позволяют максимально оперативно и точно давать прогностическую информацию, требуя минимального вмешательства со стороны оператора.

В процессе работы был создан программный комплекс (в среде DELPHI), ориентированный на решение задач анализа и прогноза временных рядов. В основу комплекса положен, описанный выше алгоритм, позволяющий создавать, и обучать многослойные искусственные нейронные сети, методом обратного распространения ошибки. В состав пакета включены инструменты, позволяющие производить спектральный анализ ряда, рассчитывать автокорреляционную функцию.

При помощи, разработанного программного пакета, проведены следующие эксперименты.

Эксперимент № 1 – Дан прогноз временного ряда, содержащего 144 значения. Результат прогноза при помощи двухслойной нейросети представлен на рис. 7. Нейросетевая модель содержит 36 нейронов входного слоя, и один нейрон выходного слоя. На рис. 7 применяются следующие обозначения: F – область прогноза; L – область обучения сети. Как видно из рисунка, нейросетевой алгоритм корректно экстраполирует кривую, сохраняя фазу и амплитуду колебаний, с учетом общего нелинейного тренда.

Количество нейронов входного слоя является главным, определяющим качество прогноза, параметром нейросетевой модели. Данный параметр целесообразно выбирать кратным одному из значимых периодов исследуемого ряда, исходя из требуемой заблаговременности прогноза. В соответствии с временным масштабом всего процесса, или же с временным масштабом одного из интересующих нас «подпроцессов».

Как показали, проведенные эксперименты, наиболее эффективное количество нейронов входного слоя (N) вычисляется по формуле: $N = k \times \mathrm{P}$, где Р – основной, значимый период процесса, определенный исходя из результата спектрального анализа ряда (рис. 8). Для каждого конкретного случая множитель *k* может варьироваться.

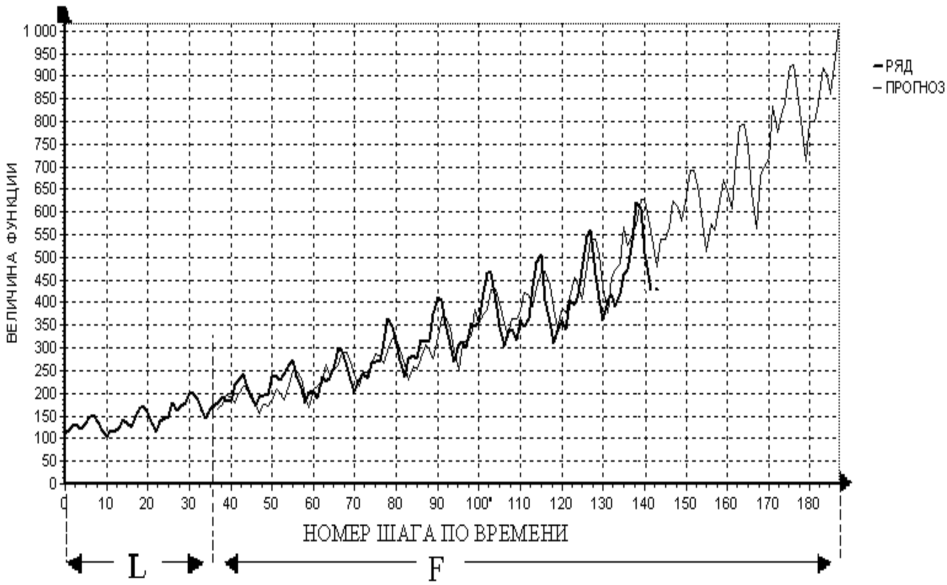

Рис. 7. Прогноз временного ряда при помощи нейросети

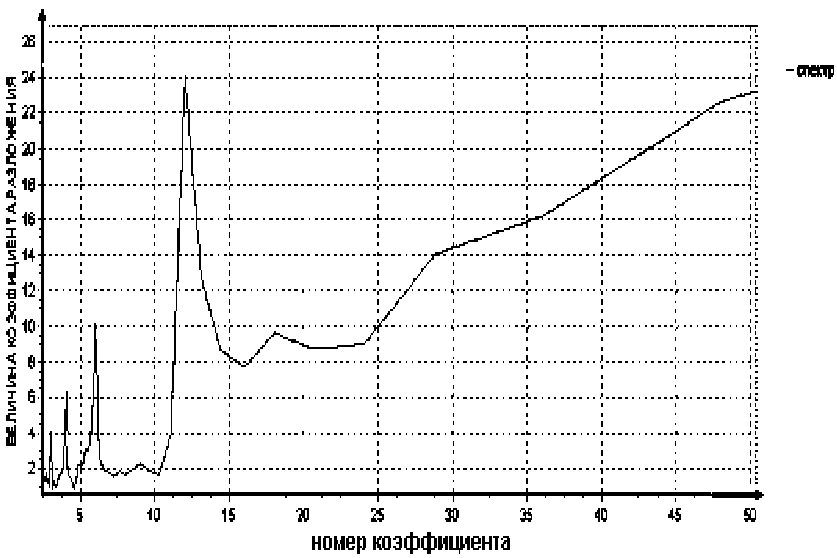

Рис. 8. Результат спектрального анализа временного ряда

Эксперимент № 2: На рис. 9 представлен результат прогноза временного ряда длиной 250 значений, при помощи трехслойной нейросети.

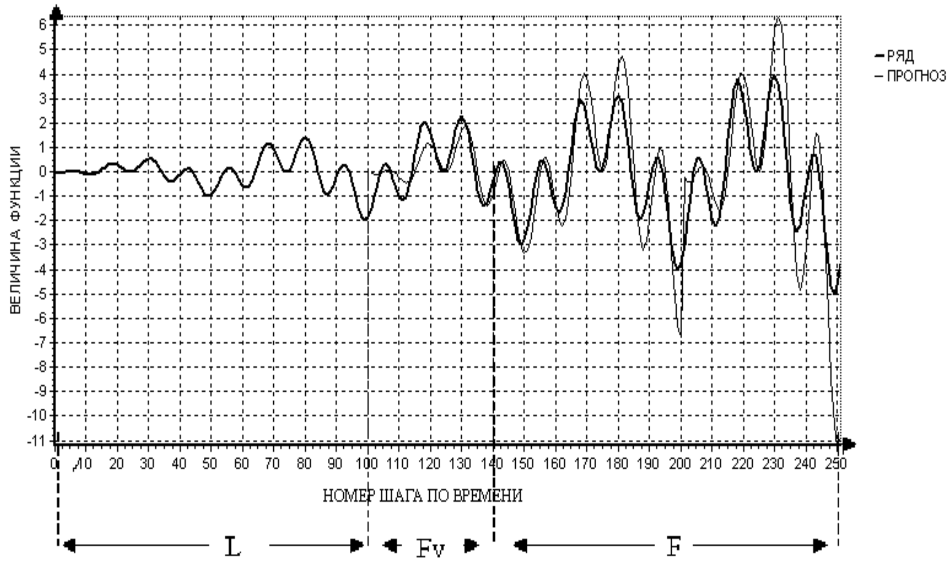

Рис. 9. Прогноз временного ряда при помощи нейросети

Нейросетевая модель содержит 100 нейронов входного слоя, 50 нейронов скрытого и один нейрон выходного слоя. В данном эксперименте модель организована более сложным образом, чем в эксперименте – 1. Временной ряд разбит на три интервала: *L* – область обучения сети (обучающее множество), *Fv* – область верификации обучения (тестовое множество), *F* – область прогноза. На рис. 10 представлен график функции *EF*, отражающий изменение среднего квадрата ошибки прогноза, по области *Fv*, в процессе обучения нейросети. Обучение сети в данном случае происходит до тех пор, пока величина функции EF, не достигнет первого локального минимума – *LM*1.

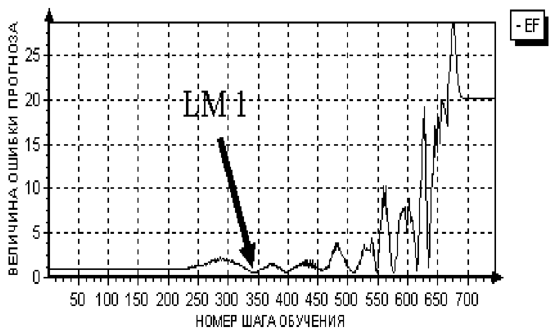

Рис. 10. График изменения среднего квадрата ошибки прогноза по области *Fv* в процессе обучения нейросети

Результат данного эксперимента еще раз подтверждает эффективность алгоритма положенного в основу модели.

Заключение

Применение описанной в работе технологии, позволяет получить нейронную сеть, допускающую простейшую реализацию и использующую минимум данных. А также, построить на ее основе модель пригодную для достаточно эффективного прогноза числовых рядов, описывающих физические процессы, проявляющие однородные колебания во времени.

Беря за основу изложенную методику, можно получить более сложные модели, путем использования многослойной сети оперирующей несколькими предикторами. То есть перейти от прогноза на основе авторегрессии, к более эффективной и интересной задаче на основе множественной регрессии.

Также в качестве возможного пути для дальнейшего развития модели, можно взять идею построения гибридной сети, включающей в себя слой самообучающихся нейронов.

Литература

1. *Поляк И.И.* Многомерные статистические модели климата. – Л.: Гидрометеоиздат, 1989, с. 29–41.
2. *Осовский С.* Нейронные сети для обработки информации / Пер. с польск. И.Д. Рудинского. – М.: Финансы и статистика, 2004. – 344 с.
3. *Уоссермен Ф.* Нейрокомпьютерная техника: Теория и практика. / Пер. на русск. яз. Ю.А. Зуев, В.А. Точенов, 1992. – 184 с. http://neuroschool.narod.ru/books/nntech.html.

Ключевые слова: прогностические модели, временные ряды, нейронные сети.