Выбор функции активации при прогнозировании нейронными сетями

Г. И. Рудой
Московский физико-технический институт, ФУПМ, каф. «Интеллектуальные системы»

Целью работы является исследование зависимости качества прогнозирования временных рядов нейронными сетями от параметров нейронной сети. В частности, анализируется зависимость от выбранной функции активации нейронов в сети, а также от параметров этой функции. Функция активации описывает выходное значение нейрона в зависимости от взвешенной суммы его входов и порогового значения срабатывания. Рассматриваются сети с прямым распространением сигналов (без обратной связи). Приводятся результаты вычислительного эксперимента по прогнозированию нейронными сетями различных временных рядов и анализируется качество прогнозов при различных функциях активации и прочих параметрах сети.

Ключевые слова: *нейронные сети с прямым распространением сигналов, линейные нейронные сети, многослойные нейронные сети, гетерогенные нейронные сети, алгоритм обратного распространения ошибки.*

Введение

Нейронная сеть с прямым распространением сигналов [2] [3] [4] — такая сеть, в которой сигнал распространяется только в одном направлении, от слоя к слою. Каждый элемент сети (нейрон) имеет один или несколько входов и один выход. Нейрон представляет собой систему из двух элементов — сумматора и функции активации. Функция активации определяет выходное значение нейрона в зависимости от результата сумматора (взвешенной суммы входов) и некоторого порогового значения. Пороговое значение также может быть представлено как еще один (неявный) вход нейрона, который не соединен ни с одним другим нейроном.

Постановка задачи

Пусть дан временной ряд $x = x(n) \mid n = 1, \ldots, m$, состоящий из некоторых числовых признаков. Требуется построить значение ряда на неизвестном промежутке, то есть, определить $x(m+1)$, $x(m+2)$ и так далее, и минимизировать среднеквадратичную ошибку прогнозирования.

Задача сводится к выбору и сравнительному анализу различных функций активации нейросети, а также различных методов обучения.

После вычисления значения в момент времени $m+1$ (и, возможно, последующие) вычисляется среднеквадратичная ошибка, являющаяся показателем качества прогнозирования, и исследуется ее зависимость от функции активации, числа нейронов, метода обучения и прочих параметров сети.

Пути решения

Совокупность известных значений временного ряда образует обучающую выборку.

Для прогнозирования временных рядов используется метод скользящего окна: выбирается p последовательных элементов, составляющих обучающую выборку и формирующих *образ*, который подаются, соответственно, на p распределительных нейронов сети. P выходных нейронов характеризуют значения функции в момент времени $p+1, p+2, \ldots, p+P$. Тогда конкретные методы прогнозирования различаются архитектурой сети, ее организацией и способом подбора и настройки весовых коэффициентов нейронной сети.

Линейная сеть

Нейронная сеть, состоящая из распределительных нейронов и одного выходного нейрона (1), имеющего линейную функцию активации, называется адаптивным нейронным элементом [5]. Выходное значение такой сети:

$$y = \sum_{i=1}^{n} w_{i1} x_i - T,$$

где T — пороговое значение указанного нейрона, а w_{j1} — весовой коэффициент, соответствующий j-ому распределительному нейрону.

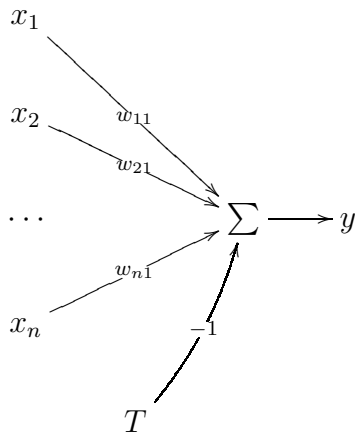

Рис. 1. Нейронная сеть из распределительных нейронов и одного нейрона с линейной функцией активации

Заметим, что нейронную сеть с несколькими выходными элементами можно представить как суперпозицию соответствующего числа нейронных сетей с одним выходным элементом, ввиду независимости выходных элементов друг от друга. В этом разделе для простоты мы будем рассматривать сети с одним выходным элементом.

Для линейной нейронной сети указанный ранее метод скользящего окна соответствует линейной авторегрессии и описывается выражением:

$$\hat{x}(n) = \sum_{k=1}^{p} w_k x(n - p + k - 1), \tag{1}$$

где $w_k, k = 1, \dots, p$ — весовые коэффициенты нейронной сети, $\hat{x}(n)$ — оценка значения ряда $x(n)$ в момент времени n. Тогда ошибка прогнозирования определяется выражением

$$e(n) = x(n) - \hat{x}(n).$$

Правило Видроу-Хоффа

Для обучения такой сети используется правило Видроу-Хоффа, известное также под названием *дельта-правила*. Оно предполагает минимизацию среднеквадратичной ошибки нейронной сети, которая для L входных образов определяется следующим образом:

$$E = \sum_{k=1}^{L} E(k) = \frac{1}{2}\sum_{k=1}^{L}(y_1^k - t^k)^2,$$

где $E(k)$ — среднеквадратичная ошибка сети для k-го образа, а y_1^k и t^k — соответственно выходное и эталонное значения нейронной сети для k-го образа.

Правило обучения Видроу-Хоффа базируется на методе градиентного спуска. Согласно этому правилу весовые коэффициенты и пороги нейронной сети необходимо изменять на $t+1$-ой итерации по следующим выражениям:

$$w_{j1}(t+1) = w_{j1}(t) - \alpha\frac{\partial E(k)}{\partial w_{j1}(t)}, \tag{2}$$

$$T(t+1) = T(t) - \alpha\frac{\partial E(k)}{\partial T(t)}, \tag{3}$$

где $j = 1, \ldots, n$, α — скорость обучения.

Производные среднеквадратичной ошибки E по данным параметрам:

$$\frac{\partial E}{\partial w_{j1}(t)} = (y_1^k - t^k)x_i^k, \tag{4}$$

$$\frac{\partial E}{\partial T(t)} = -(y_1^k - t^k), \tag{5}$$

где x_i^k — j-ая компонента k-го образа.

Подставляя (4) и (5) в (2) и (3), получаем выражения для обучения нейронной сети по дельта-правилу:

$$w_{j1}(t+1) = w_{j1}(t) - \alpha(y_1^k - t^k)x_i^k, \tag{6}$$

$$T(t+1) = T(t) + \alpha(y_1^k - t^k). \tag{7}$$

Б. Видроу и М. Хофф доказали [5], что данный закон всегда позволяет находить весовые коэффициенты нейронного элемента таким образом, чтобы минимизировать среднеквадратичную ошибку сети независимо от начальных значений весовых коэффициентов.

Алгоритм обучения, основанный на дельта-правиле, состоит из следующих шагов:

1. Задается скорость обучения α $(0 < \alpha < 1)$ и минимальная среднеквадратичная ошибка сети E_m, которую необходимо достичь в процессе обучения.
2. Случайным образом инициализируются весовые коэффициенты и порог нейронной сети.
3. Подаются входные образы на нейронную сеть и вычисляются выходные значения сети.
4. Осуществляется изменение весовых коэффициентов согласно (6) и (7).
5. Алгоритм работает до тех пор, пока суммарная среднеквадратичная ошибка не станет меньше заданной $(E \leqslant E_m)$.

Использование псевдообратной матрицы

Для настройки весовых коэффициентов линейной нейронной сети с целью минимизации среднеквадратичной ошибки можно использовать матричное решение системы линейных уравнений.

Пусть размерность обучающей выборки — L, число выходных нейронов — m, число входных нейронов — n. Тогда матрицы выходных значений, входных значений и весовых коэффициентов, соответственно, имеют вид:

$$Y = \begin{bmatrix} y_1^1 & y_1^2 & \dots & y_1^L \\ y_2^1 & y_2^2 & \dots & y_2^L \\ \dots & \dots & \dots & \dots \\ y_m^1 & y_m^2 & \dots & y_m^L \end{bmatrix} \text{— матрица выходных значений,}$$

$$X = \begin{bmatrix} x_1^1 & x_1^2 & \dots & x_1^L \\ x_2^1 & x_2^2 & \dots & x_2^L \\ \dots & \dots & \dots & \dots \\ x_n^1 & x_n^2 & \dots & x_n^L \end{bmatrix} \text{— матрица входных значений,}$$

$$W = \begin{bmatrix} w_{11} & w_{12} & \dots & w_{1m} \\ w_{21} & w_{22} & \dots & w_{2m} \\ \dots & \dots & \dots & \dots \\ w_{n1} & w_{n2} & \dots & w_{nm} \end{bmatrix} \text{— матрица весовых коэффициентов.}$$

Тогда уравнение линейной нейронной сети можно представить в виде:

$$Y = XW$$

В случае использования порогов нейронных элементов в матрицу X добавляется строка, содержащая значения -1, а в матрицу W — строка, в которой находятся пороговые значения нейронных элементов T.

В работе [6] показано, что наилучшим приближенным решением системы линейных уравнений, при котором среднеквадратичная ошибка сети достигает своего наименьшего значения, является выражение

$$W = X^+ Y,$$

где X^+ — псевдообратная матрица для матрицы X, определяемая следующим образом:

$$X^+ = (X^T X)^{-1} X^T$$

Это решение единственно [6] и минимизирует среднеквадратичную ошибку сети: $E = \| Y - WX \|^2$.

Существуют различные эффективные способы нахождения псевдообратной матрицы [6].

Заметим, что при использовании псевдообратной матрицы возникают проблемы, когда матрица $X^T X$ является вырожденной.

Многослойная нейронная сеть

Архитектура многослойной нейронной сети (рис. 2) состоит из множества слоев нейронных элементов. *Входной слой* нейронных элементов выполняет распределительные функции, *выходной слой* служит для обработки информации от предыдущих слоев и выдачи результата. Слои, расположенные между входным и выходным слоями, называются *промежуточными* или *скрытыми*. И выходной, и скрытые слои являются обрабатывающими. Выход каждого нейронного элемента предыдущего слоя нейронной сети соединен со всеми входами нейронных элементов следующего слоя.

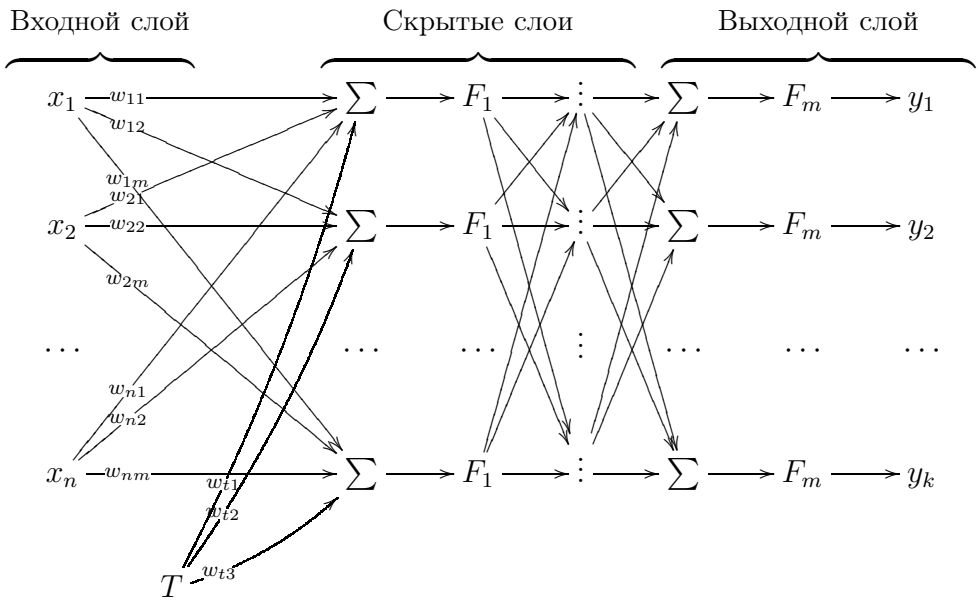

Рис. 2. Нейронная сеть с $m-1$ скрытыми слоями, n входами и k выходами

В качестве функции активации нейронных элементов обычно используются гиперболический тангенс или сигмоидная функция. Если соответствующие элементы имеют одинаковую функцию активации, то сеть называется *гомогенной*, иначе — *гетерогенной*.

Пусть $W^{(i)}$ — матрица весовых коэффициентов i-го слоя. Тогда, например, для гомогенной нейронной сети с двумя скрытыми слоями матрица выходных значений определяется как

$$Y = F_3(F_2(F_1(XW^{(1)})W^{(2)})W^3),$$

где $X = (x_1, x_2, \ldots, x_n)$ — вектор-строка входных сигналов, F_i — определяемый функцией активации оператор нелинейного преобразования для i-го слоя. Для гомогенной нейронной сети $F_i = F_j \mid \forall i, j$, для гетерогенной сети некоторые (либо все) из операторов могут быть различны.

Число слоев в многослойной нейронной сети определяет, каким образом входное пространство образов может быть разбито на подпространства меньшей размерности [2]. Так, двухслойная нейронная сеть с одним слоем нелинейных нейронов разбивает входное пространство образов на классы при помощи гиперплоскости [7]. Трехслойная нейронная сеть, где в качестве двух последних слоев используются нейронные элементы с нелинейной функцией активации, позволяет формировать любые выпуклые области в пространстве решений [7] [8]. Четырехслойная нейронная сеть с тремя нелинейными слоями дает возможность получать область решений любой формы и сложности, в том числе и невыпуклой.

А. Н. Колмогоров показал [9], что любую непрерывную функцию n переменных на единичном отрезке $[0; 1]$ можно представить в виде суммы конечного числа одномерных функций:

$$f(x_1, x_2, \ldots, x_n) = \sum_{p=1}^{2n+1} g\left(\sum_{i=1}^{n} \lambda_i \varphi_p(x_i)\right),$$

где функции g и φ_p являются одномерными и непрерывными, $\lambda_i = const$ для всех i.

Данная теорема легла в основу построения многослойных нейронных сетей для аппроксимации функций. Из нее следует, что любую непрерывную функцию $f : [0;1]^n \to [0;1]$ можно аппроксимировать при помощи трехслойной нейронной сети, имеющей n входных, $2n+1$ скрытых и один выходной нейрон. Данный результат был затем обобщен на многослойную сеть с алгоритмом обратного распространения ошибки [10], [3], [11].

Алгоритм обратного распространения ошибки

Алгоритм обратного распространения ошибки был предложен в [12] и является эффективным методом для обучения нейронных сетей. Данный алгоритм минимизирует среднеквадратичную ошибку нейронной сети. Покажем основные идеи, использованные при построении этого алгоритма [2].

Согласно методу градиентного спуска изменение весовых коэффициентов и порогов нейронной сети происходит по следующему правилу:

$$w_{ij}(t+1) = w_{ij}(t) - \alpha \frac{\partial E}{\partial w_{ij}(t)} \tag{8}$$

$$T_j(t+1) = T_j(t) - \alpha \frac{\partial E}{\partial T_j(t)}, \tag{9}$$

где $E = \frac{1}{2}\sum_j (y_j - t_j)^2$ — среднеквадратичная ошибка нейронной сети для одного образа.

В [2] показано, что

$$\frac{\partial E}{\partial w_{ki}} = \gamma_i F'(S_i)\gamma_k \tag{10}$$

$$\frac{\partial E}{\partial T_i} = -\gamma_i F'(S_i) \tag{11}$$

где γ_i — выходное значение i-го нейрона.

Подставляя (10) и (11) в (8) и (9), получаем следующие выражения, показывающие, что для минимизации среднеквадратичной ошибки сети весовые коэффициенты и пороги нейронных элементов должны изменяться следующим образом:

$$w_{ij}(t+1) = w_{ij}(t) - \alpha\gamma_j F'(S_j) y_i \tag{12}$$

$$T_j(t+1) = T_j(t) + \alpha\gamma_i F'(S_j) \tag{13}$$

Эти выражения определяют правило обучения многослойных нейронных сетей в общем виде, которое называется *обобщенным дельта-правилом.*

Определим выражения (12) и (13) для типичных функций активации нейронных элементов. Примем обозначение $S_j = \sum_i w_{ij} y_i - T_j$.

— Для *сигмоидной функции* выходное значение j-го элемента определяется следующим образом:

$$y_j = \frac{1}{1 + e^{-S_j}}$$

В результате обобщенное дельта-правило можно представить в виде:

$$w_{ij}(t+1) = w_{ij}(t) - \alpha\gamma_j y_j(1-y_j)y_i \tag{14}$$

$$T_j(t+1) = T_j + \alpha\gamma_j y_j(1-y_j)$$

Ошибка для j-го нейрона выходного слоя:

$$\gamma_j = y_j - t_j,$$

для j-го нейрона скрытого слоя

$$\gamma_j = \sum_{i=1}^{m} \gamma_i y_i(1-y_i)w_{ji},$$

где m — число нейронов следующего слоя по отношению к слою j.

— Для *биполярной сигмоидной функции* выходное значение j-го элемента определяется по формуле

$$y_j = \frac{2}{1+e^{-S_j}} - 1$$

Обобщенное дельта-правило:

$$w_{ij}(t+1) = w_{ij}(t) - \frac{1}{2}\alpha\gamma_j(1-y_j^2)y_i$$

$$T_j(t+1) = T_j(t) + \frac{1}{2}\alpha\gamma_j(1-y_j^2)$$

Ошибка для j-го нейрона выходного и скрытого слоев, соответственно:

$$\gamma_j = y_j - t_j$$

$$\gamma_j = \frac{1}{2}\sum_i \gamma_i(1-y_i^2)w_{ij}$$

— Для *гиперболического тангенса* выходное значение j-го нейрона определяется как

$$y_j = \mathrm{th}(S_j) = \frac{e^{S_j} - e^{-S_j}}{e^{S_j} + e^{-S_j}}$$

Производная этой функции имеет вид $F'(S_j) = 1-y_j^2$, поэтому правило обучения можно представить в виде:

$$w_{ij}(t+1) = w_{ij}(t) - \alpha\gamma_j(1-y_j^2)y_i \tag{15}$$

$$T_j(t+1) = T_j(t) + \alpha\gamma_j(1-y_j^2) \tag{16}$$

Ошибка для j-го нейрона выходного и скрытого слоев соответственно:

$$\gamma_j = y_j - t_j$$

$$\gamma_j = \sum_i \gamma_i(1-y_i^2)w_{ij}$$

Сам алгоритм обратного распространения представляет следующую последовательность шагов.

1. Задаются шаг обучения α $(0 < \alpha < 1)$ и желаемая среднеквадратичная ошибка нейронной сети E_m.
2. Инициализируются случайным образом весовые коэффициенты и пороговые значения нейронной сети.
3. Подаются последовательно образы из обучающей выборки на вход нейронной сети. При этом для каждого входного образа выполняются следующие действия:
   (а) Производится фаза прямого распространения образа по сети, при этом вычисляется выходная активность всех нейронных элементов сети:
   $$y_j = F(\sum_i w_{ij} y_i - T_j),$$
   где индекс j характеризует нейроны следующего слоя по отношению к слою i.
   (б) Осуществляется фаза обратного распространения сигнала, в результате которой определяется ошибка γ_j нейронных элементов для всех слоев сети. При этом, соответственно, для выходного и скрытого слоев:
   $$\gamma_j = y_j - t_j$$
   $$\gamma_j = \sum_i \gamma_i F'(S_i) w_{ji}$$
   В последнем выражении i характеризует нейронные элементы следующего слоя по отношению к слою j.
   (в) Происходит изменение весовых коэффициентов и порогов нейронных элементов для каждого слоя нейронной сети в соответствии с (12) и (13).
4. Вычисляется суммарная среднеквадратичная ошибка нейронной сети E:
   $$E = \frac{1}{2} \sum_{k=1}^{L} \sum_j (y_j^k - t_j^k)^2,$$
   где L — размерность обучающей выборки.
5. Если $E > E_m$, то происходит переход к шагу 3, иначе алгоритм заканчивается.

Таким образом, алгоритм функционирует до тех пор, пока суммарная среднеквадратичная ошибка сети не станет меньше заданной.

Алгоритм обратного распространения ошибок наделен следующими проблемами:

— Неизвестность выбора числа слоев и количества нейронных элементов в слое.
— Медленная сходимость градиентного метода с постоянным шагом обучения.
— Сложность выбора скорости обучения α. Слишком малая скорость увеличивает время обучения и приводит к нахождению лишь локального минимума, в то время как большая скорость может привести к расхождению процесса.
— Невозможность определения точек локального и глобального минимумов, так как градиентный метод их не различает.
— Влияние случайной инициализации весовых коэффициентов на поиск минимума (неустойчивость).

Заметим, что важную роль играет порядок величин случайно инициализируемых синаптических связей [13] [4]. Так, для сигмоидной функции активации нейронных элементов, при больших по модулю значениях весовых коэффициентов выходная активность будет близка к единице или нулю, и тогда значение выражения $y_j(1-y_j)$ будет близко к нулю, и, согласно (14), весовые коэффициенты будут изменяться незначительно. Это приведет к тому, что процесс обучения остановится в ближайшем локальном минимуме от стартовой точки. В [13] рекомендуется случайно выбирать значения весовых коэффициентов, имеющих порядок $w_{ij} \approx \frac{1}{\sqrt{n(i)}}$, где $n(i)$ — число элементов в слое i. Другие авторы рекомендуют инициализировать весовые коэффициенты случайными числами в диапазоне порядка $[-0.05; 0.05]$.

Другим важным вопросом является число нейронных элементов в скрытых слоях. С одной стороны, при возрастании их числа растет точность, с другой, при слишком большой размерности скрытых слоев возникает явление *перетренировки сети*, ухудшающее обобщающие способности нейронных сетей. Поэтому число нейронных элементов в скрытом слое должно быть меньше числа тренировочных образцов.

Для нейтрализации застревания метода градиентного спуска в нежелательных минимумах применяется метод тяжелого шарика [13]. В этом случае модификация синаптических связей нейронной сети происходит в соответствии с выражением

$$\Delta w_{ij}(t+1) = -\alpha\gamma_j F'(S_j) y_i + \xi \Delta w_{ij}(t),$$

где ξ — постоянная величина, называемая *моментным параметром*. Значение моментного параметра выбирается из диапазона $[0; 1]$, на практике часто используется значение $\xi = 0.9$ [13].

Алгоритм послойного обучения

Как отмечалось ранее, алгоритм обратного распространения ошибки имеет ряд недостатков, среди которых зависимость от начальной инициализации синаптических связей. Это происходит из-за наличия локальных минимумов в целевой функции, и в результате не всякая попытка приводит к обучению. В данном разделе приводится альтернативный алгоритм обучения сети, проявляющий устойчивость при решении некоторых задач.

Рассмотрим алгоритм на примере нейронной сети с одним скрытым слоем, где в качестве функции активации используется гиперболический тангенс. Вводя адаптивный шаг и используя алгоритм обратного распространения ошибки, выражения для обучения нейронной сети можно представить следующим образом:

$$w_{ij}(t+1) = w_{ij}(t) - \alpha(t)\gamma_j(1-y_j^2)y_i \tag{17}$$

$$T_j(t+1) = T_j(t) + \alpha(t)\gamma_j(1-y_j^2), \tag{18}$$

где

$$\alpha(t) = \frac{\sum_j \gamma_j^2(1-y_j^2)}{(1+\sum_i y_i^2)\sum_j \gamma_j^2(1-y_j^2)}$$

Алгоритм послойного обучения предполагает также проводить модификацию выходов нейронных элементов скрытых слоев. Такая модификация осуществляется в соответствии с методом градиентного спуска:

$$y_j(t+1) = y_j(t) - \alpha(t)\frac{\partial E}{\partial y_j(t)}$$

Здесь $\alpha(t)$ находится из метода наискорейшего спуска:

$$\alpha(t) = \min\left\{E(y_j(t) - \alpha(t)\frac{\partial E}{\partial y_j(t)})\right\}$$

Показывается [2], что оптимальное значение адаптивного шага обучения:

$$\alpha = \frac{\sum_j (y_i - t_j)\sum_i \gamma_i w_{ij}}{\sum_j (\sum_i \gamma_i w_{ij})^2}$$

Тогда выходы нейронных элементов скрытого слоя должны модифицироваться на основе выражения:

$$y_i(t+1) = y_i(t) - \alpha\gamma_i \tag{19}$$

Тогда сам алгоритм послойного обучения состоит из следующих шагов:

1. Случайная инициализация весовых коэффициентов нейронной сети и задание минимальной среднеквадратичной ошибки E_m.
2. На вход сети последовательно подаются L входных образов, и для последнего слоя весовые коэффициенты и пороги модифицируются в соответствии с (17) и (18).
3. На вход сети последовательно подаются L входных образов, и выходные значения y_i нейронов скрытого слоя модифицируются в соответствии с (19).
4. Шаги 2 и 3 повторяются до тех пор, пока суммарная среднеквадратичная ошибка сети не станет меньше, чем заданная.
5. Для L входных образов модифицируются весовые коэффициенты и пороги следующего слоя нейронной сети. При этом ошибка i-го нейронного элемента $\overline{\gamma_i} = y_i - \overline{y_i}$.
6. Процедура повторяется с шага 2, пока суммарна я среднеквадратичная ошибка сети не станет меньше заданной.

Гетерогенные нейронные сети

Рассмотрим простейшую гетерогенную сеть, состоящую из одного скрытого слоя с нелинейной функцией активации нейронных элементов и выходного линейного нейрона. Тогда выходное значение сети:

$$y = \sum_i \nu_i y_i - T, \tag{20}$$

где ν_i — i-ый весовой коэффициент выходного нейрона, y_i — выходные значения нейронных элементов скрытого слоя:

$$y_i = F(S_i) = F(\sum_l w_{li}x_l - T) \tag{21}$$

Для различных слоев нейронной сети необходимо использовать разные выражения для вычисления адаптивного шага обучения. Адаптивный шаг для выходного слоя вычисляется по формуле:

$$\alpha_1 = \frac{1}{1 + \sum_i y_i^2} \tag{22}$$

Адаптивный шаг для скрытого слоя:

$$\alpha_2 = \frac{\sum_i C_i \sum_j (y_j - t_j) w_{ij}}{F_1'(0) F_2'(0) \sum_j (\sum_i C_i w_{ij})^2} \tag{23}$$

Показывается [2], что обучающие правила для выходного слоя и для скрытого слоя соответственно записываются как:

$$v_i(t+1) = v_i(t) - \alpha_1(t)(y - t)y_i \tag{24}$$

$$T(t+1) = T(t) + \alpha_1(t)(y - t) \tag{25}$$

$$w_{li}(t+1) = w_{lj}(t) - \alpha_2(t)\gamma_i F'(S_i)x_l \tag{26}$$

$$T_i(t+1) = T_i(t) + \alpha_2(t)\gamma_i F'(S_i) \tag{27}$$

Для обучения может использоваться *алгоритм многократного распространения ошибки* в связи с нестабильностью процесса обучения из-за использования различных функций активации.

Алгоритм обратного распространения ошибки предполагает для каждого тренировочного набора модификацию синаптических связей всех слоев нейронной сети. При этом изменение весовых коэффициентов одного слоя нейронной сети происходит без учета изменения остальных слоев. Это может привести к нестабильности процесса обучения, который характеризуется отсутствием тенденции к снижению среднеквадратичной ошибки сети. В гетерогенных сетях поэтому может возникнуть разсинхронизация процесса обучения между разными слоями сети. Алгоритм многократного распространения ошибки, в свою очередь, предполагает на каждой итерации модификацию синаптических связей только одного слоя нейронной сети. В соответствии с этим каждый образ будет последовательно подаваться на нейронную сеть столько раз, сколько настраиваемых слоев имеет сеть.

Пусть p — число настраиваемых слоев нейронной сети. Тогда алгоритм многократного распространения ошибки состоит из следующих шагов:

1. Задается желаемая среднеквадратичная ошибка E_m.
2. Синаптические связи инициализируются случайным образом.
3. Счетчик числа настраиваемых слоев инициализируется числом p.
4. Подается первый тренировочный набор на вход нейронной сети. Производится фаза прямого и обратного распространения сигнала. В результате осуществляется модификация весовых коэффициентов и порогов нейронных элементов только для p-го слоя нейронной сети:

$$w_{ip}(t+1) = w_{ip}(t) - \alpha_2(t)\gamma_p F_p'(S_p)y_i$$

$$T_p(t+1) = T_p(t) + \alpha(t)\gamma_p F_p'(S_p),$$

где $i = p - 1$.

5. Счетчик декрементируется.
6. Если $p \neq 0$, перейти к шагу 4, иначе перейти к шагу 7.
7. Повторяется процесс обучения, начиная с шага 3, для всех тренированных наборов обучающей выборки.
8. Вычисляется суммарная среднеквадратичная ошибка E, и если $E > E_m$, происходит переход к шагу 3, иначе процесс обучения завершается.

Логарифмическая функция активации

Рассмотрим такой нейрон, выходное значение которого представляется в виде выражения ($S_i = \sum_l w_{li}x_l - T$):

$$y_i = \ln(S_i + \sqrt{S_i^2 + 1}) \tag{28}$$

Такой нейрон имеет логарифмическую функцию активации, возрастающую монотонно от $(-\infty; \infty)$ и имеющую точку перегиба в начале координат.

Рассмотрим гетерогенную нейронную сеть, имеющую скрытый слой нейронов с логарифмической функцией активации и один линейный выходной нейрон. Такая нейронная сеть используется для решения задач прогнозирования.

Выражения для модификации настраиваемых параметров сети можно представить в следующем виде:

$$w_{li}(t+1) = w_{li}(t) - \alpha_2(t)(y - t)y_i\nu_i x_l y_i^l$$

$$T_i(t+1) = T_i(t) + \alpha_2(t)(y - t)y_i\nu_i y_i^l$$

При этом

$$\alpha_2 = \frac{\sum_i \nu_i^2 y_i}{(1 + \sum_l x_l^2)\sum_i \nu_i^2 (y_i')^2}$$

Обучение происходит согласно выведенным ранее формулам для обучения гетерогенных сетей (24), (25) и (22).

Численный эксперимент

В численном эксперименте нейронные сети с различными архитектурами и конфигурациями тестируются для выявления оптимальных конфигураций параметров и характерных особенностей поведения. Используются как синтетические данные, так и реальные:

— Прогнозирование функции

$$y = 0.1\sin(3\mathbf{x}) + 0.5 \tag{29}$$

— Прогнозирование функции

$$y = 10\sin\mathbf{x} + 5\sin(3\mathbf{x}) \tag{30}$$

— Прогнозирование функции

$$y = 10\sin\mathbf{x} + 5\sin(3\mathbf{x}) + 2\sin(30\mathbf{x}) + \sin(50\mathbf{x}) \tag{31}$$

— Данные по потреблению электроэнергии в Турции.

В ходе численного эксперимента сначала задавался размер одной выборки данных, эффективно определяющий количество входов нейронной сети, и размер части выборки, используемой для обучения. Также задавалось количество точек, которые необходимо спрогнозировать. Нейронная сеть обучалась до достижения заданной среднеквадратичной ошибки на обучающей выборке, затем производился эксперимент по прогнозированию данных на P шагов вперед и вычислялась среднеквадратичная ошибка на спрогнозированных данных. Если архитектура нейронной сети подразумевала одновременное вычисление всех P точек, то они вычислялись сразу, в противном случае вычислялась лишь одна или несколько следующих точек, они добавлялись в вектор уже известных данных и снова подавались на вход нейронной сети.

Линейная сеть с правилом обучения Видроу-Хоффа

Линейная сеть при обучении до среднеквадратичной ошибки в $E_m = 0.001$ демонстрирует хорошие результаты при прогнозировании периодических функций как на короткий промежуток времени, так и на несколько периодов. Однако, при помощи линейной сети, обучаемой по правилу Видроу-Хоффа, удалось достичь требуемой E_m только для первых двух периодических функций (29) и (30). Для функции (31) метод либо не сходился, либо за критическое число итераций, установленное в 10^5, не достигалась требуемая точность, и процесс прерывался.

Из 3 видно, что среднеквадратичная ошибка прогнозирования как функция от числа нейронов (и, то есть, от числа входных данных) имеет один или несколько явно выраженных минимумов: при недостаточном числе нейронов нейронная сеть не может точно спрогнозировать следующие точки, а при избыточном числе наступает переобучение. Также из этого графика видно, что если предсказывание осуществляется на основе тех точек, которые уже были в обучающей выборке (как в случае 25 точек, для зеленого графика), то среднеквадратичная ошибка существенно меньше, чем в обратном случае (для 20 точек, красная линия).

На графике 4 приведены результаты прогнозирования для двух нейронных сетей: с 4 и 14 входными нейронами соответственно. Видно, что прогноз нейронной сети из 4 элементов практически совпадает с эталоном, в то время как для прогноза нейронной сети с 14 входными элементами прогноз существенно отклоняется от реальных значений. Среднеквадратичные ошибки равны 0.0452432 и 0.2324168 соответственно — то есть, для нейронной сети из 14 элементов наблюдается переобучение. Аналогичная картина наблюдается на графике 5, на котором представлены результаты прогнозирования зашумленного синуса (30) нейросетями с 5 и 15 входными нейронами.

Вид зависимости среднеквадратичной ошибки от числа итераций при прогнозировании (30) указан на 14. Заметим дополнительно, что для сети с 15 входными элементами потребовалось 2303 итерации для достижения требуемой среднеквадратичной ошибки в 0.001, а для сети с 5 нейронами — 1520 итераций.

Заметим, что данным методом не удалось предсказать (31) и реальные данные: заданная точность не достигалась за установленное предельное число итераций при достаточно малой скорости обучения α, либо метод расходился с увеличением α.

Линейная сеть с использованием псевдообратной матрицы

При использовании псевдообратной матрицы следует заботиться, чтобы $L - p - P \geqslant k$, где L — размерность выборки, p — число входов, k — зависящее от ряда целое число, например, 2 для (29), 3 для (30) и $6 - 7$ для (31), иначе метод перестает работать.

На графике 5 приведены прогнозы сетью с 8 входными нейронами на 10 и 11 точек вперед соответственно. В первом случае прогноз совпадает с реальными значениями, в то время как во втором случае наглядно демонстрируется ошибочный прогноз, возникающий из-за недостаточной размерности выборки относительно желаемого числа предсказанных точек и входных нейронов.

Графики 7 и 9 демонстрируют возможности по предсказанию реальных временных рядов. Заметим, что среднеквадратичная ошибка, деленная на число предсказанных точек, на три порядка меньше абсолютных значений, и приблизительно в 50 раз меньше локальных разбросов значений. График 11 показывает, что нейронная сеть не в состоянии предсказать случайные девиации: в реальных данных на этом участке случился небольшой провал, который не был предсказан нейронной сетью. График 13 показывает, что нейронная сеть успешно справляется с прогнозированием на число точек, сопоставимое с размерностью обучающей выборки.

Итак, метод демонстрирует хорошую точность предсказания и высокую применимость даже при недостаточной размерности входных данных относительно желаемого числа входов и размерности предсказываемых данных: достаточно, чтобы соблюдалось вышеупомянутое условие. Кроме того, при наличии эффективных методов расчета псевдообратной матрицы метод соответственным образом гораздо быстрее метода, использующего правило обучения Видроу-Хоффа. Заметим так, что этим методом удалось эффективно предсказать (31) и реальные данные, что было невозможно при обучении по правилу Видроу-Хоффа. Заметим так же, что для продолжения временного ряда из реальных данных по 5000-7000 точкам на 500 точек вперед, пришлось увеличить размер стека в Scilab до 10 миллионов слов двойной точности, а для продолжения на 2200 точек по 4000 точкам с размерностью обучающей выборки 6500 — до 50 миллионов.

Гомогенная многослойная сеть

Анализировались гомогенные нейронные сети с одним скрытым слоем и различным числом нейронов в входном, выходном и скрытом слоях, а также различными функциями активации. Для обучения во всех случаях использовался алгоритм обратного распространения ошибки.

При использовании биполярной сигмоидной функции в качестве функции активации ни в одном из случаев не удалось достигнуть среднеквадратичной ошибки в 0.001 за число итераций, не превышающее 10^5, поэтому все дальнейшие экспериментальные данные приведены для нейронной сети, использующей гиперболический тангенс как функцию активации. Заметим, что, в связи с ограниченной областью значений гиперболического тангенса, необходимо масштабировать обучающую выборку к соответствующему диапазону значений.

График 14 демонстрирует характерный вид зависимости среднеквадратичной ошибки от номера итерации.

Качество прогнозирования зависит также от числа нейронов в скрытом слое, которое определялось как $k \lfloor p + P \rfloor$, где k варьировалось в ходе эксперимента. Графики зависимости приведены на 15. По оси абсцисс отложено число нейронов во входном слое, разные линии соответствуют разным числам k, которые приведены на легенде. На 16 те же графики представлены как трехмерная поверхность для наглядности. Видно, что, начиная с $k = 2$, изменение числа нейронов в скрытом слое практически не влияет на качество прогнозирования.

Также стоит отметить, что с ростом числа нейронов во входном слое одновременно растет среднеквадратичная ошибка прогнозирования и возрастает число итераций, необходимых для достижения заданной среднеквадратичной ошибки на обучающей выборке, поэтому в реальных приложениях имеет смысл ограничиваться меньшим числом нейронов во входном слое. Однако, при слишком малом числе входных нейронов сеть демонстрирует недостаточную обобщающую способность (17, приведены графики для сетей с 3 и 4 нейронами во входном слое).

С задачей прогнозирования функции (31) сеть справляется, в отличие от однослойной сети, обучаемой по правилу Видроу-Хоффа, но имеет существенно большую среднеквадратичную ошибку, чем однослойная же сеть, но обучаемая методом псевдообратной матрицы (18).

Заключение

В работе рассмотрены различные варианты архитектур нейронных сетей без обратной связи, протестированы на синтетических и реальных данных различные сети с различными методами обучения, проанализирована зависимость качества прогнозирования и скорости обучения от параметров сетей.

В частности:

— Наблюдается зависимость качества прогнозирования от размера окна, с одним или несколькими минимумами, и ухудшением качества с чрезмерным ростом размера окна (явление переобучения).
— Метод Видроу-Хоффа для обучения однослойной линейной нейронной сети перестает работать для достаточно сложных зависимостей.
— Метод обучения однослойной линейной нейронной сети при помощи псевдообратной матрицы позволяет быстро (за время, необходимое для вычисления псевдообратной матрицы) получить минимально возможную ошибку прогнозирования на данных.
— Сходимость алгоритма обратного распространения ошибки обеспечивается лишь при использовании гиперболического тангенса.
— Существует оптимальное значение входного числа нейронов для каждого типа временных рядов.
— Повышение сложности нейронной сети путем введения дополнительных нейронов в скрытый слой не является эффективным методом повышения качества прогнозирования для всех типов временных рядов.

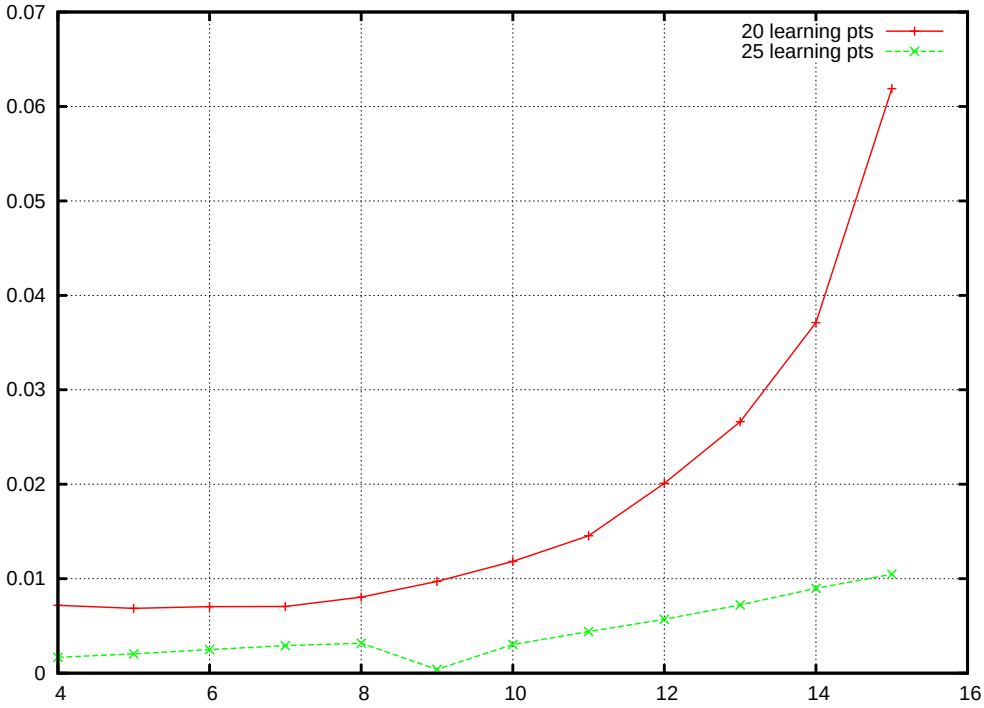

Рис. 3. Зависимость среднеквадратичной ошибки прогнозирования от числа входных нейронов при прогнозировании (29)

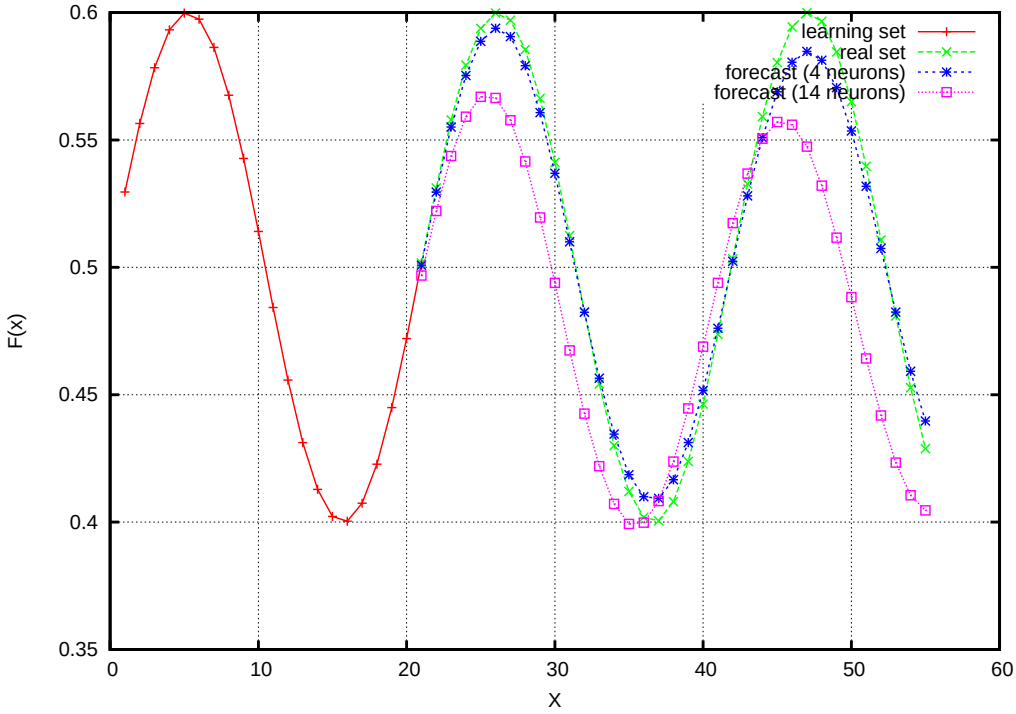

Рис. 4. Прогноз для функции (29) с 20 точками обучающей выборки: сети с 4 и 14 входными нейронами и 35 предсказанными точками, обучение по правилу Видроу-Хоффа

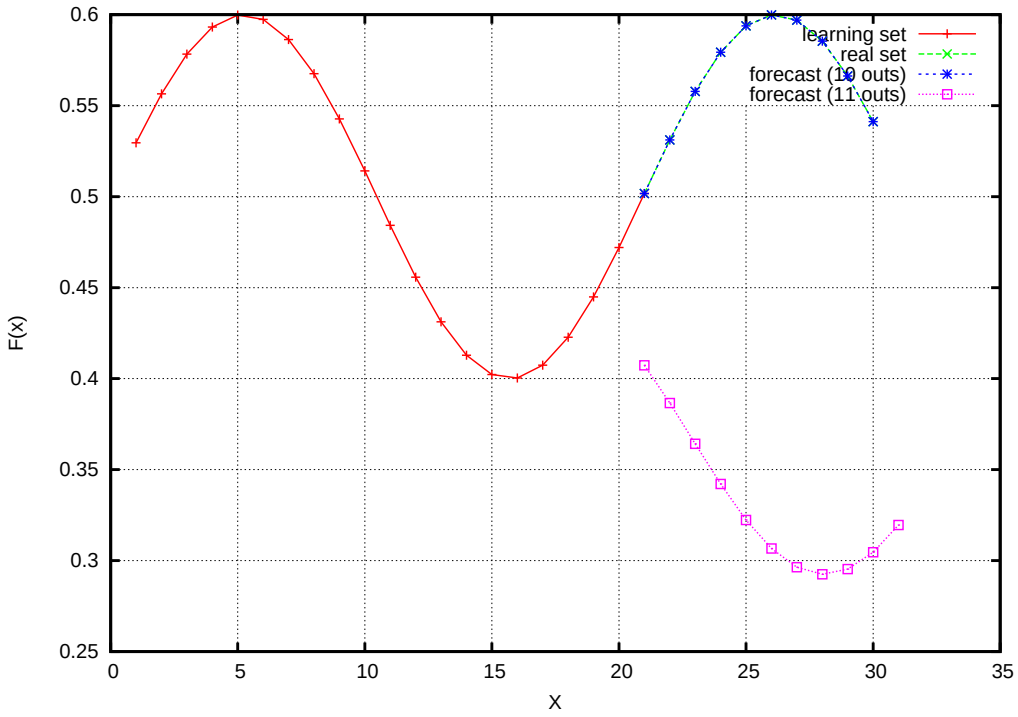

Рис. 5. Прогноз для функции (29) с 20 точками обучающей выборки: сети с 8 входными нейронами и 10 и 11 предсказанными точками, обучение методом псевдообратной матрицы

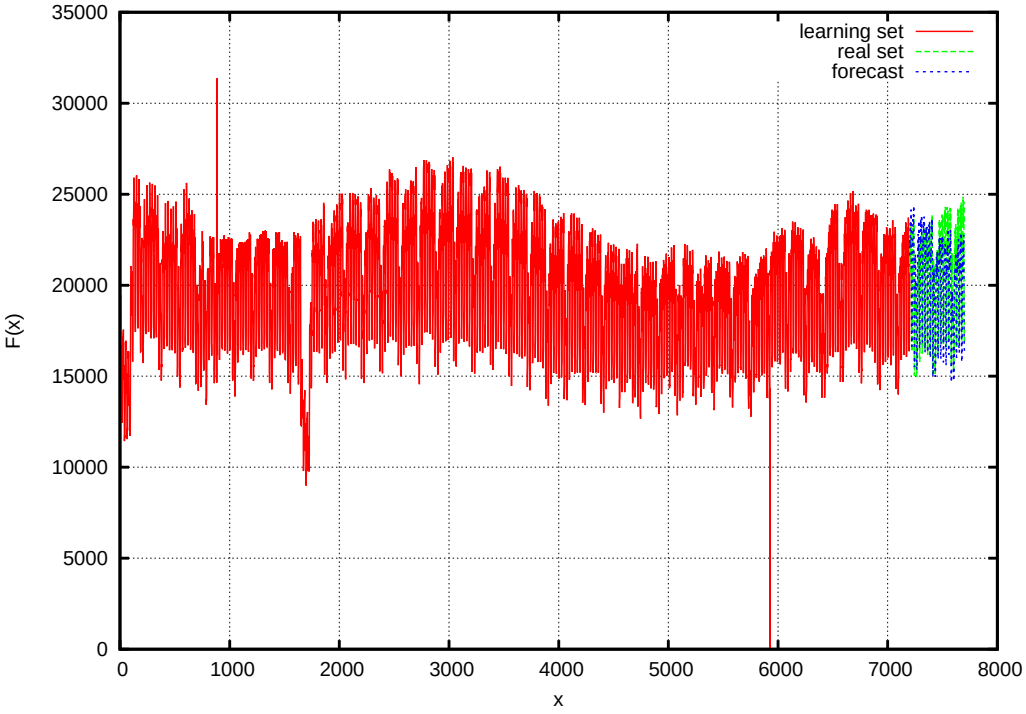

Рис. 6. Предсказанные значения для реальных данных, 7200 точек в обучающей выборке, 6600 входов, предсказание на 500 точек

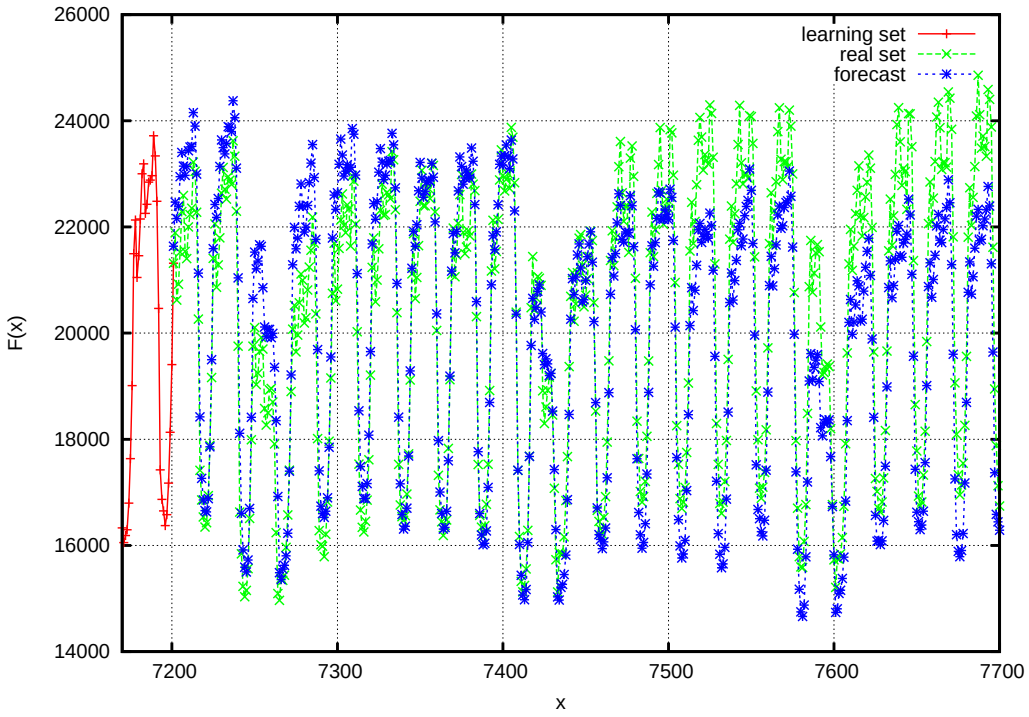

Рис. 7. Предсказанные значения для реальных данных, 7200 точек в обучающей выборке, 6600 входов, предсказание на 500 точек, увеличенная предсказанная часть

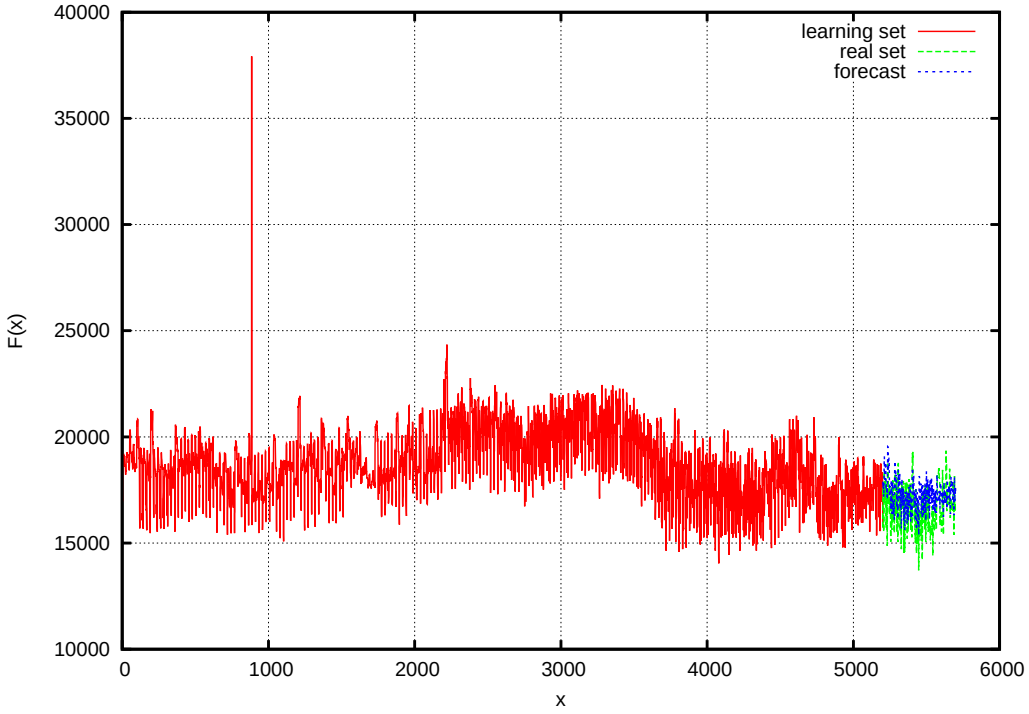

Рис. 8. Предсказанные значения для реальных данных, 5200 точек в обучающей выборке, 4600 входов, предсказание на 500 точек

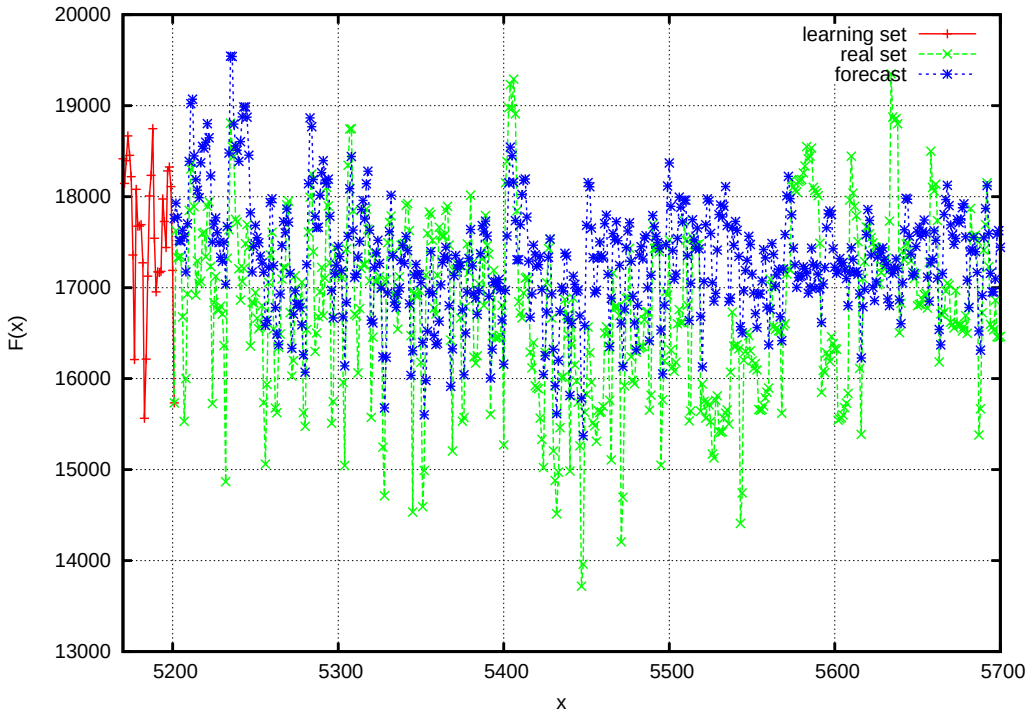

Рис. 9. Предсказанные значения для реальных данных, 5200 точек в обучающей выборке, 4600 входов, предсказание на 500 точек, увеличенная предсказанная часть

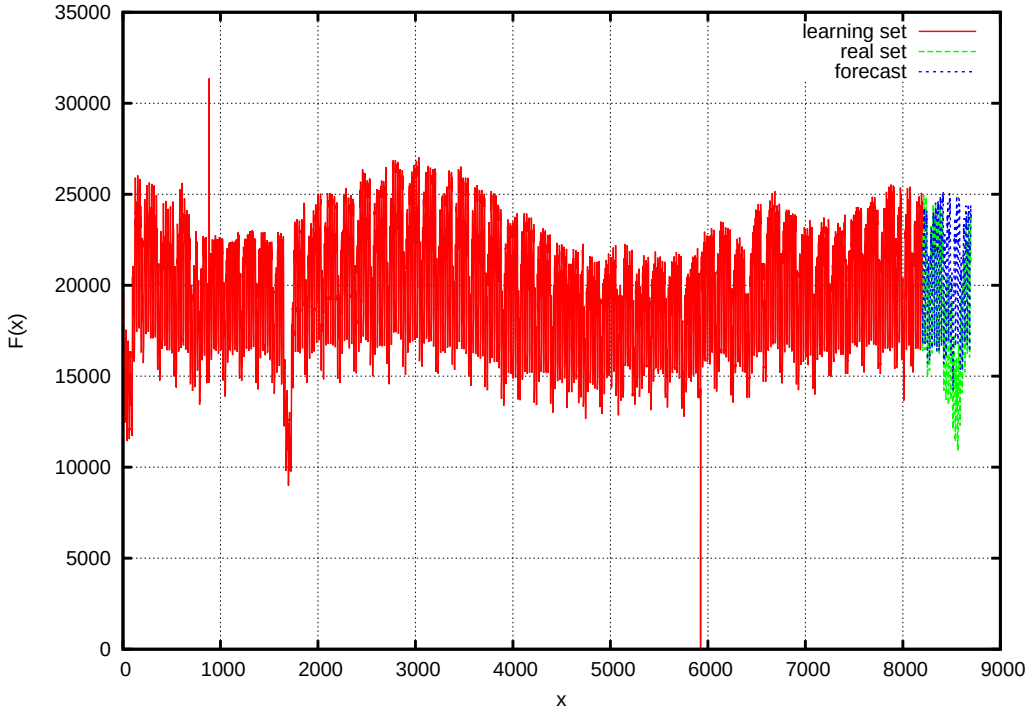

Рис. 10. Предсказанные значения для реальных данных, 8200 точек в обучающей выборке, 7600 входов, предсказание на 500 точек

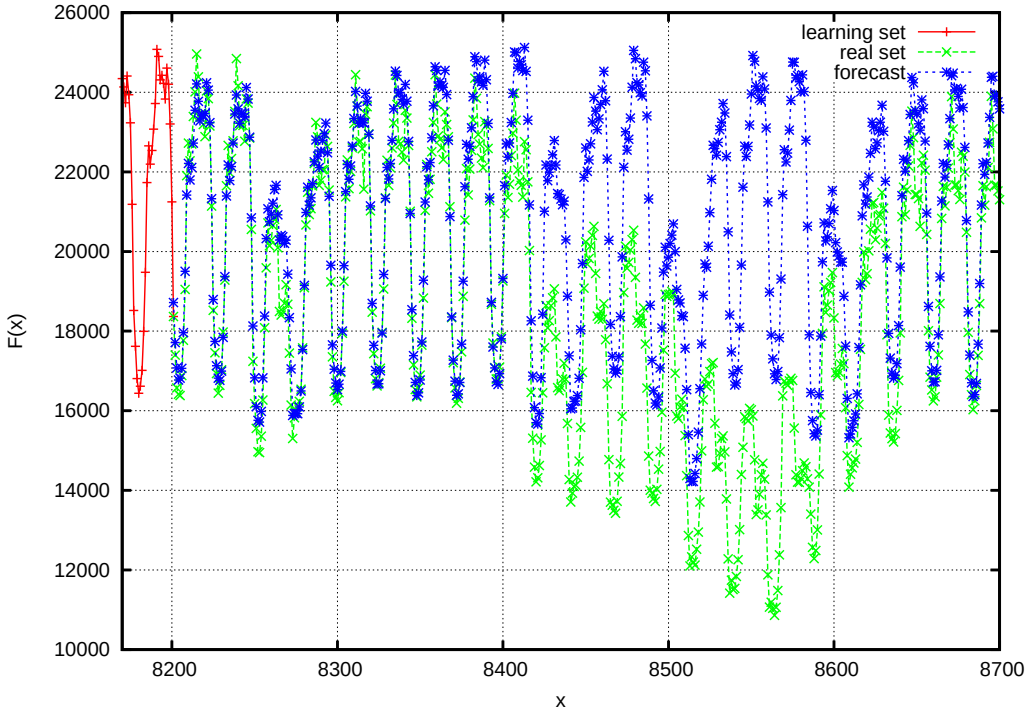

Рис. 11. Предсказанные значения для реальных данных, 8200 точек в обучающей выборке, 7600 входов, предсказание на 500 точек, увеличенная предсказанная часть

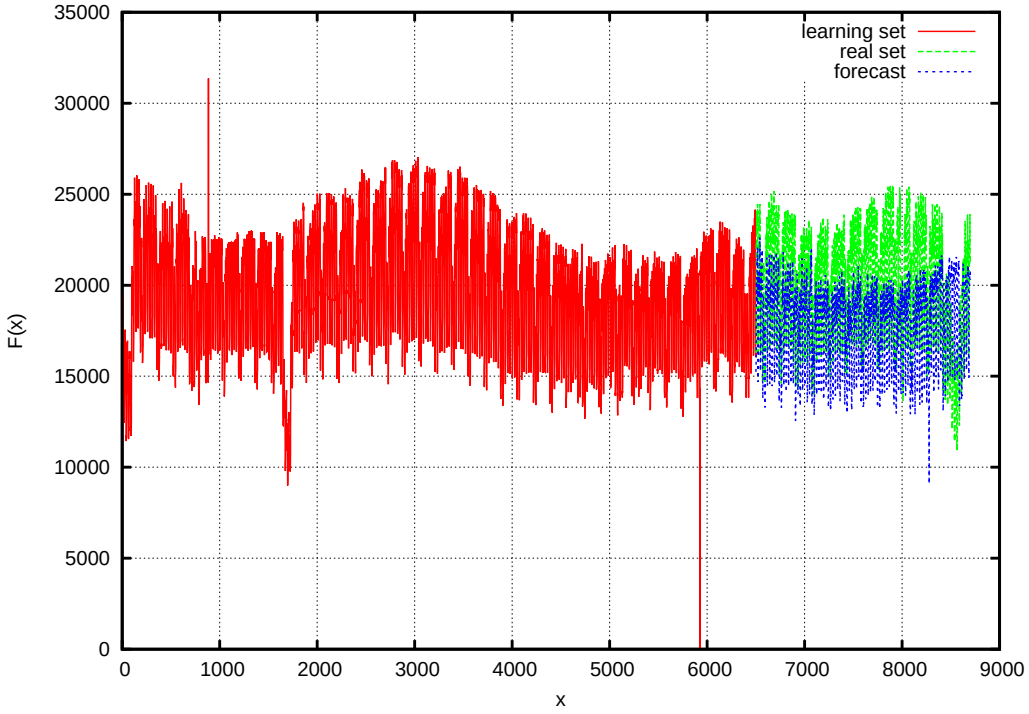

Рис. 12. Предсказанные значения для реальных данных, 6500 точек в обучающей выборке, 4000 входов, предсказание на 2200 точек

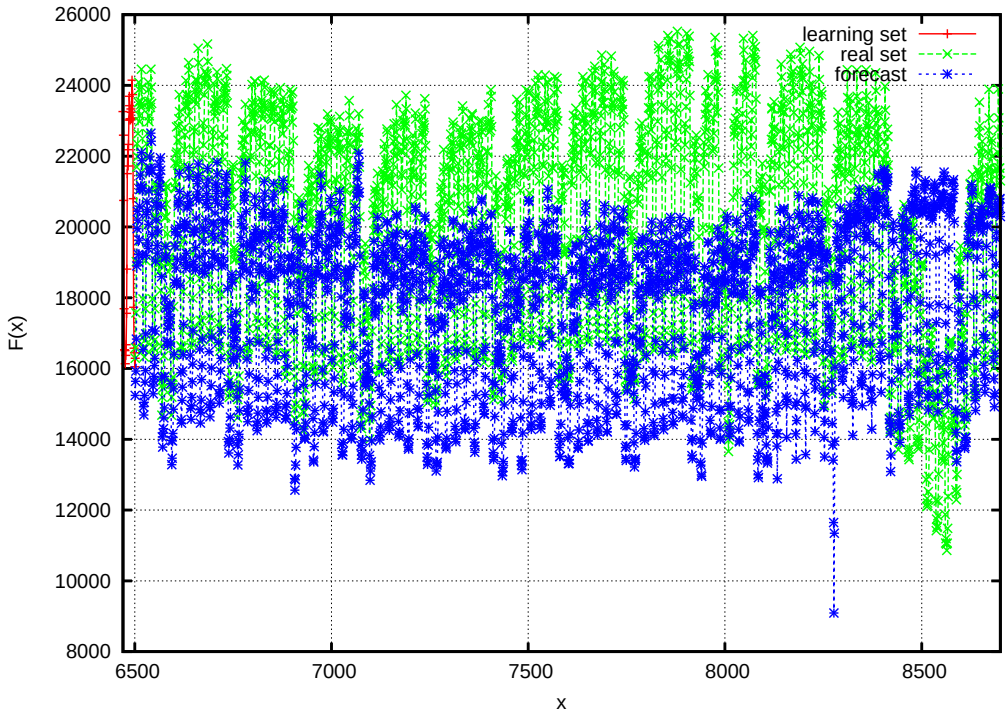

Рис. 13. Предсказанные значения для реальных данных, 6500 точек в обучающей выборке, 4000 входов, предсказание на 2200 точек, увеличенная предсказанная часть

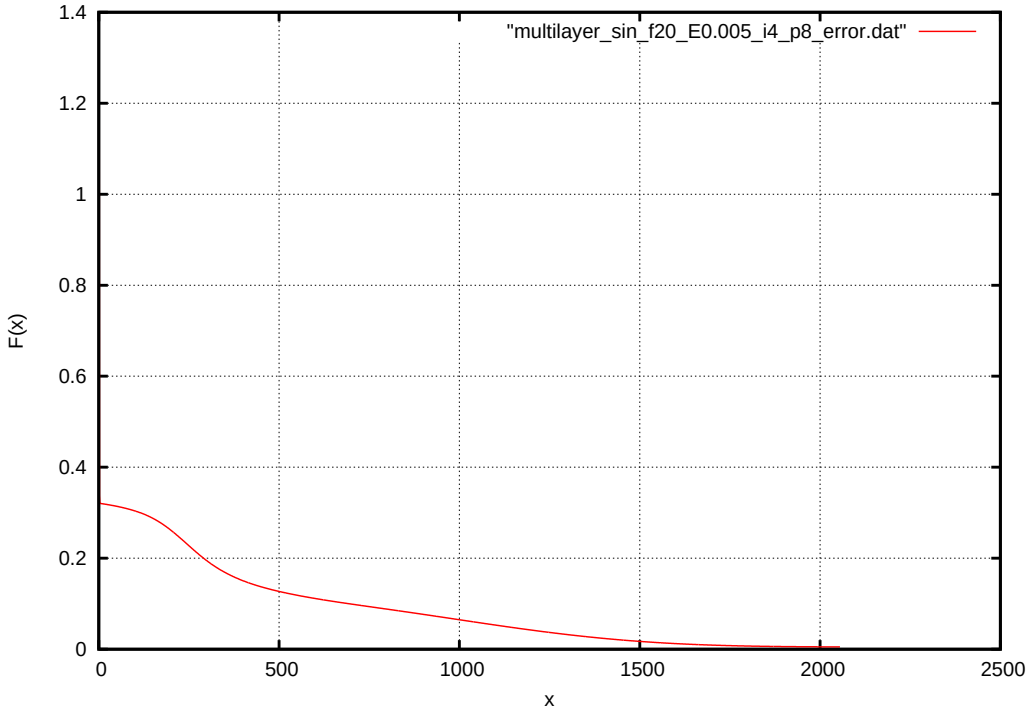

Рис. 14. График убывания ошибки на обучающей выборке для функции (29) с 20 точками обучающей выборки и 8 предсказанными точками

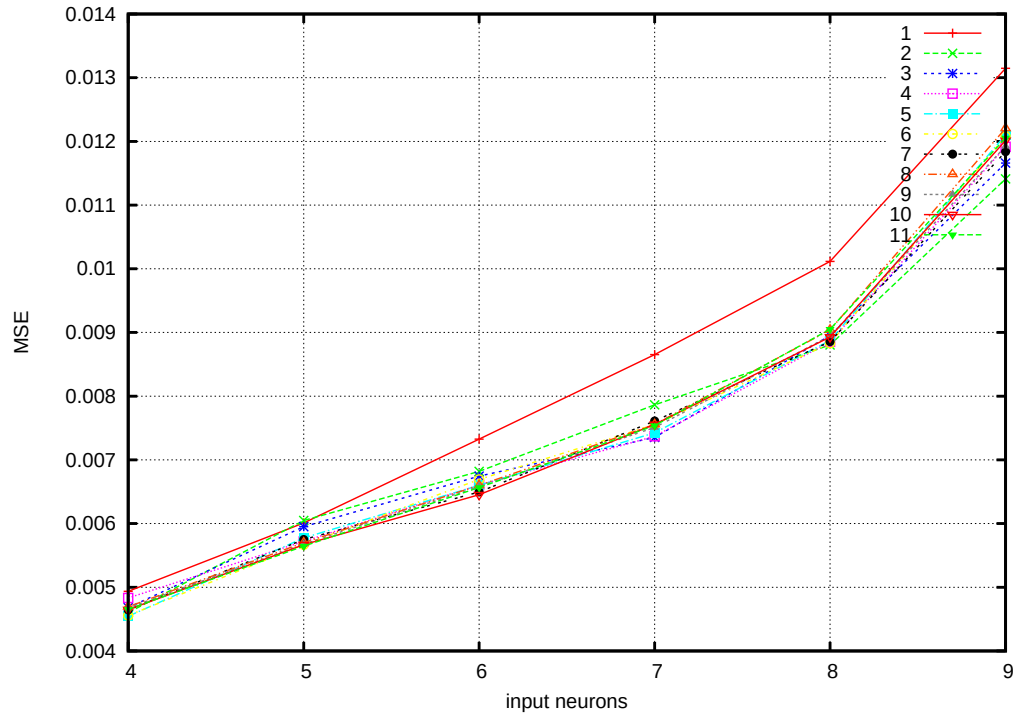

Рис. 15. Зависимость приведенной среднеквадратичной ошибки прогнозирования функции (29) от числа нейронов во входном и скрытом слоях

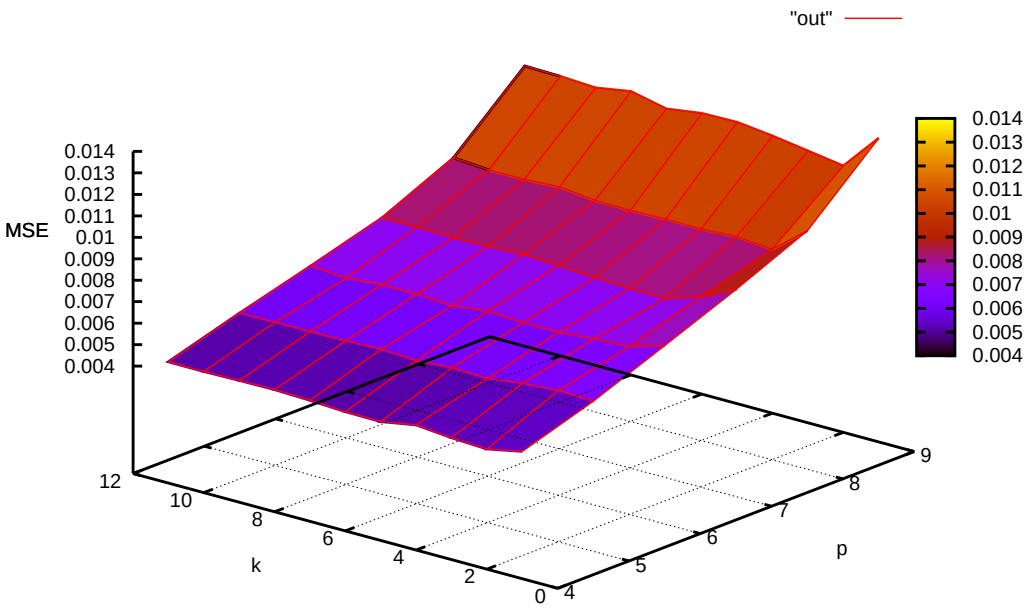

Рис. 16. Зависимость приведенной среднеквадратичной ошибки прогнозирования функции (29) от числа нейронов во входном и скрытом слоях (трехмерная визуализация)

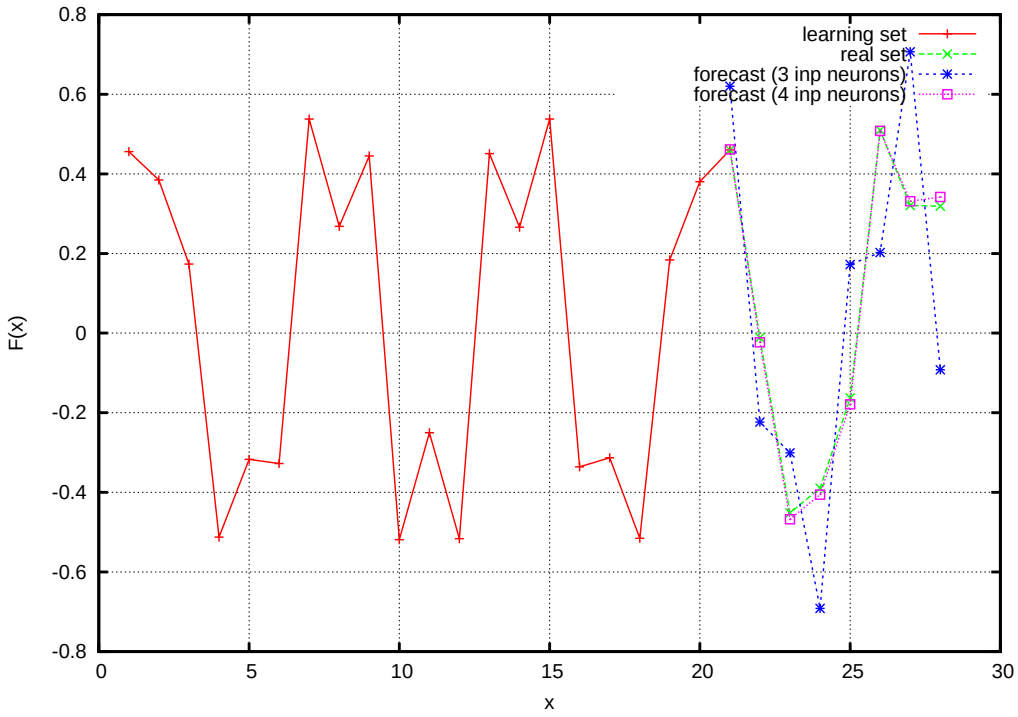

Рис. 17. Прогноз для функции (30) с 20 точками обучающей выборки: многослойные сети с 3 и 4 входными нейронами

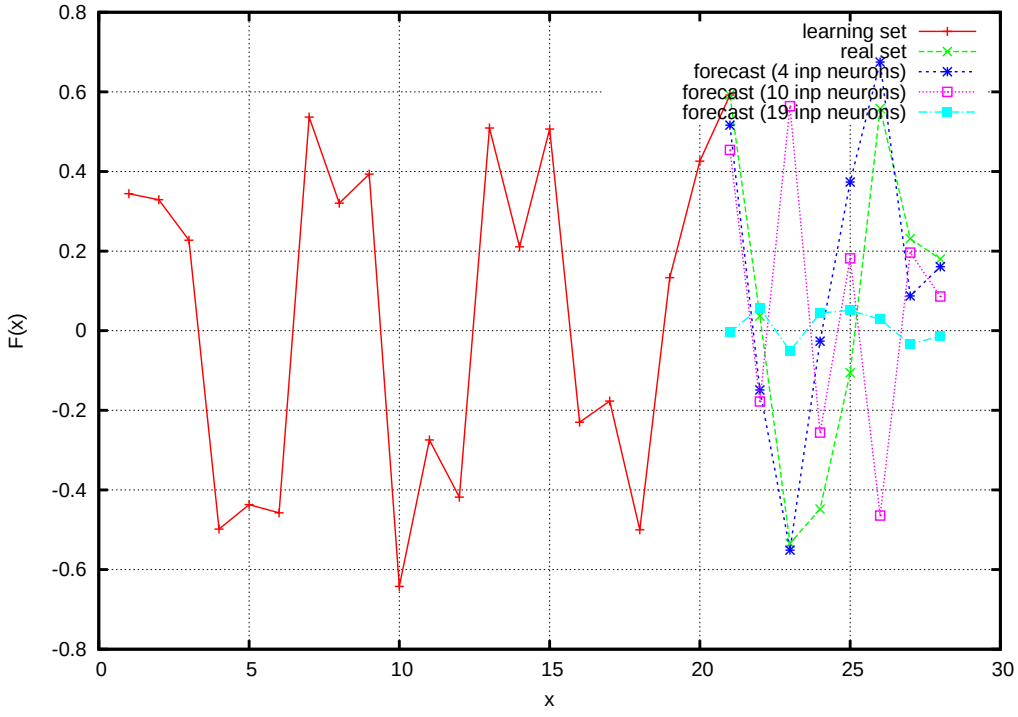

Рис. 18. Прогноз для функции (31) с 20 точками обучающей выборки: многослойные сети с 4, 10 и 19 входными нейронами

Литература

[1] В. Н.Солнцев, Д. Л. Данилов, А. А. Жиглявский. *Главные Компоненты Временных Рядов: Метод "Гусеница"*, С.-Петербургский государственный университет, 1997.

[2] В.А. Головко. Нейронные сети обучение, организация и применение. 2001.

[3] K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989.

[4] Ben J. A. Kroese and P. Patrick van der Smagt. An introduction to neural networks. 1993.

[5] B. Widrow and M. E. Hoff. Adaptive switching circuits. pages 96–104, 1960.

[6] Ф. Гантмахер. Теория матриц. 1988.

[7] Richard P. Lippmann. An introduction to computing with neural nets. *IEEE ASSP Magazine*, April 1987.

[8] А. Н. Скурихин. Нейронные сети: определения, концепции, применение. 1991.

[9] А. Н. Колмогоров. Представление непрерывных функций многих переменных суперпозицией функций одной переменной и сложением. *ДАН*, 5:953–956, 1958.

[10] Raul Rojas. Theorie der neuronalen netze: Eine systematische einfuehrung. 1996. 4., korrigierter Nachdruck.

[11] T. Maxwell, C. L. Giles, Y. C. Lee, and H. H. Chen. Nonlinear dynamics of artificial neural systems. 1986.

[12] D. Rumelhart, G. Hinton, and R. Williams. Learning representations by back-propagating errors. *Nature*, 323:533–536, October 1986.

[13] J. Hertz, A. Krogh, and R. Palmer. Introduction to the theory of neural computation. 1991.