

Исследование сходимости при прогнозировании нейронными сетями с обратной связью

Н. П. Балдин

baldin01@gmail.com

Исследуется зависимость скорости сходимости при прогнозировании временных рядов от параметров нейронной сети с обратной связью. В качестве модели нейронной сети используется сеть Джордана. Предлагается проанализировать скорость сходимости в зависимости от выбора функции активации (сигмоидной, гиперболического тангенса), от числа нейронов в промежуточном слое и от ширины скользящего окна. Также разбирается способ повышения скорости сходимости при использовании обобщенного дельта-правила.

Ключевые слова: машинное обучение, нейронные сети, сеть Джордана, градиентный спуск, обобщенное дельта-правило.

Введение

Первые шаги в области нейронных сетей были сделаны В. Мак-Каллохом (W. McCulloch) и В. Питсом (W. Pitts). Они показали, что при помощи нейронных элементов можно реализовать исчисление любых логических функций [3]. В 1949 г. Хейбб (D. Hebb) предложил правило обучения, которое стало математической основой для обучения ряда нейронных сетей. В 1957—1962 гг. Ф. Розенблат (F. Rosenblatt) предложил и исследовал модель нейронной сети, которую он назвал персептроном. Она была популярной до 1969 г., когда М. Минский (M. Minsky) и С. Пайперт (S. Papert) опубликовали монографию [4], в которой были доказаны ограниченные возможности персептрона. В 1986 г. ряд авторов (D. Rumelhart, G. Hinton, R. Williams) предложили алгоритм обратного распространения ошибки для обучения многослойной нейронной сети [5].

Рекуррентные сети — нейронные сети, в которых выходы нейронных элементов последующих слоев имеют соединение с нейронами предшествующих слоев (Рис. 1). Выходное значение нейрона промежуточного слоя:

$$S_i(t) = \sum_{j=1}^n w_{ij}(t)x_j + \sum_{k=1}^p u_{ki}(t)y_k(t-1) - T_i,$$

где w_{ij} — весовой коэффициент между j -м нейроном входного и i -м нейроном промежуточного слоев; p — число нейронов выходного слоя; u_{ki} — весовой коэффициент между k -м контекстным нейроном и i -м нейроном промежуточного слоя; T_j — пороговое значение i -го нейрона промежуточного слоя; n — размерность входного вектора. Это приводит к возможности учета результатов преобразования нейронной сетью информации на предыдущем этапе для обработки входного вектора на следующем этапе функционирования сети. Существует множество архитектур нейронных сетей [7, 6, 2, 1]. Для изучения сходимости в данной работе используется простейшая архитектура рекуррентной сети — сеть Джордана, представленная в [1]. В качестве модельных данных используются тригонометрические функции (периодический, зашумленный и аperiodический временные ряды).

Постановка задачи

Дано: $f_i : X \rightarrow R, i = 1, \dots, n$ — числовые признаки, $\langle \mathbf{w}, \mathbf{x}_j \rangle = \sum_{i=1}^n f_i(x)w_i - w_0$ — линейная комбинация признаков $w_0, w_1, \dots, w_n \in R$.

Определение 1. Нейрон — вершина взвешенного ориентированного графа (нейронной сети).

Определение 2. Функция активации — оператор нелинейного преобразования входных данных нейрона промежуточного слоя.

Совокупность известных значений временного ряда образуют обучающую выборку. Обозначим ее размерность L . Архитектура нейронной сети: n входных нейронов (в дальнейшем, ширина окна), p нейронов промежуточного слоя, один выходной нейрон, характеризующий значение в момент времени $n + 1$. Требуется по известным значениям ряда определить значение в последующий момент времени.

Вычисляется среднеквадратичная ошибка.

$$E = \sum_i^m (y_i - t_i)^2,$$

где y_i — значение на i -м выходе, t_i — эталонное значение. Затем запускается алгоритм обратного распространения ошибки. Исследуется сходимость от числа нейронов в промежуточном слое, от функции активации, и ширины окна. Для этого нужно для каждого набора параметров найти число итераций алгоритма. Далее найдем оптимальный набор параметров для каждого временного ряда. Проанализируем один из способов ускорения сходимости.

Решение задачи

Описание архитектуры. Выходное значение j -го нейрона последнего слоя определяется по формуле

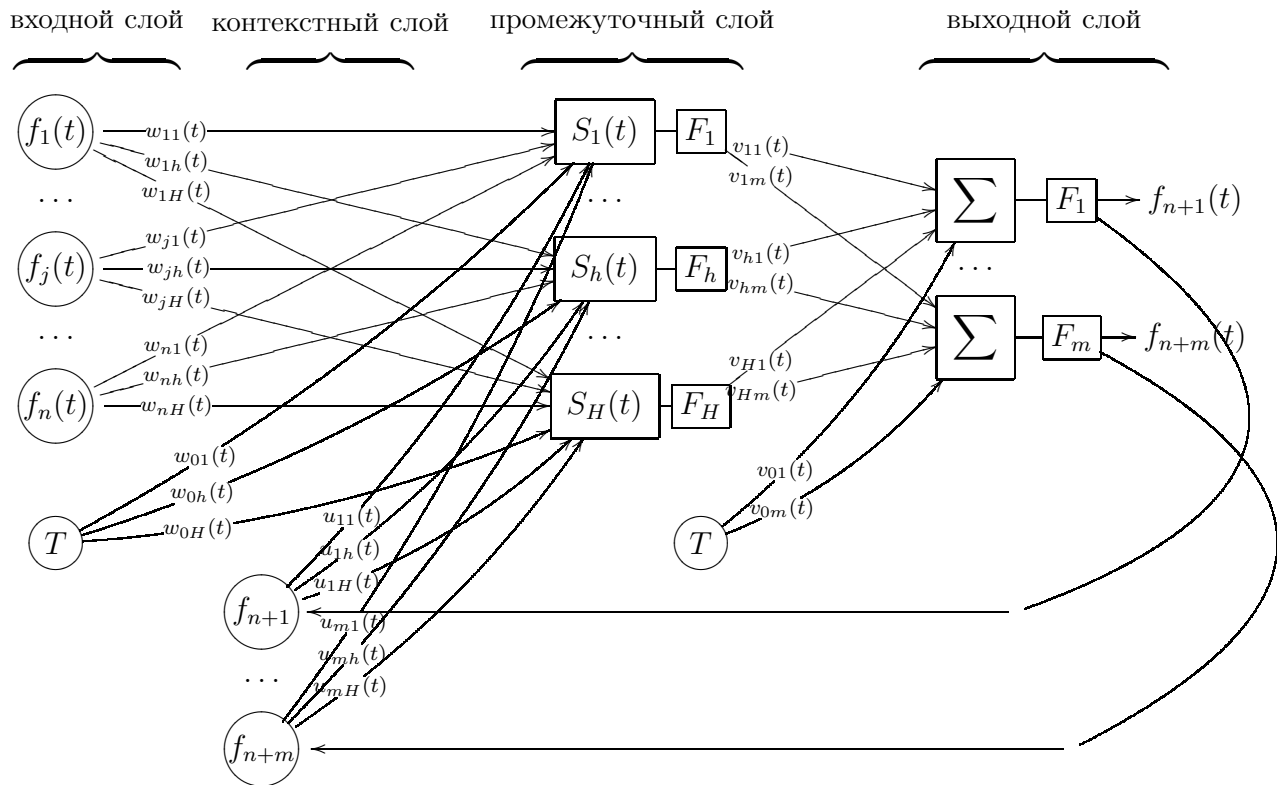


Рис. 1. Архитектура сети Джордана

$$y_j(t) = \sum_{i=1}^m v_{ij} p_i(t) - T_j,$$

где v_{ij} —весовой коэффициент между i -м нейроном промежуточного слоя и j -м нейроном выходного слоев. $p_i(t)$ — выходное значение i -го нейрона промежуточного слоя; T_j — пороговое значение j -го нейрона выходного слоя.

Взвешенная сумма i -го нейрона промежуточного слоя определяется следующим образом:

$$S_i(t) = \sum_{j=1}^n w_{ij}(t)x_j + \sum_{k=1}^p u_{ki}(t)y_k(t-1) - T_i.$$

Тогда выходное значение i -го нейрона промежуточного слоя $p_i(t) = F(S_i(t))$, где F — функция активации. В качестве F используется гиперболический тангенс или сигмоидальная функция.

Алгоритм обучения. Алгоритм обратного распространения ошибок был предложен в [5] и является эффективным средством для обучения многослойных нейронных сетей. Алгоритм обучения рекуррентной сети в общем случае состоит из следующих шагов:

1. В начальный момент времени $t = 1$ все контекстные нейроны устанавливаются в нулевое состояние, т. е. их выходные значения равняются нулю.
2. Входной образ подается на сеть и происходит прямое распространение его в нейронной сети.
3. В соответствии с алгоритмом обратного распространения ошибки модифицируются весовые коэффициенты и пороговые значения нейронов.
4. Устанавливается $t = t + 1$ и осуществляется переход к шагу 2.

Определяется среднеквадратичная ошибка нейронной сети:

$$E = \frac{1}{2} \sum_j (y_j - t_j)^2,$$

где t_j — эталонное выходное значение j -го нейрона. Согласно методу градиентного спуска изменение весовых коэффициентов и порогов нейронной сети происходит по следующему правилу:

$$\begin{aligned} w_{ij}(t+1) &= w_{ij}(t) - a \frac{\partial E}{\partial w_{ij}(t)}, \\ T_j(t+1) &= T_j(t) - a \frac{\partial E}{\partial T_j(t)}. \end{aligned}$$

Ошибка j -го выходного слоя:

$$h_j = y_j - t_j.$$

Теорема 1. Для любого промежуточного слоя i ошибка i -го нейрона определяется рекурсивным образом через ошибки нейронов следующего слоя j :

$$h_i = \sum_{j=1}^m h_j F'(S_j) w_{ij},$$

где m — число нейронов следующего слоя по отношению к слою i ; w_{ij} — вес между i -м и j -м нейронами различных слоев; S_j — взвешенная сумма j -го нейрона.

Используя результаты данной теоремы, сформулированной и доказанной в [1], можно определить ошибки нейронов промежуточного слоя через ошибки нейронов следующего слоя по отношению к промежуточному слою.

Теорема 2. Производные среднеквадратичной ошибки по весовым коэффициентам и порогам нейронов для любых двух слоев i и j многослойной сети определяются следующим образом:

$$\begin{aligned}\frac{\partial E}{\partial w_{ij}} &= -h_j F'(S_j) y_j; \\ \frac{\partial E}{\partial T_j} &= h_j F'(S_j).\end{aligned}$$

Следствие 1. Для минимизации среднеквадратичной ошибки сети весовые коэффициенты и пороги нейронов должны изменяться с течением времени следующим образом

$$\begin{aligned}w_{ij}(t+1) &= w_{ij}(t) - a h_j F'(S_j) y_j; \\ T_j(t+1) &= t_j(t) + a h_j F'(S_j),\end{aligned}$$

где a - скорость обучения. Доказательства можно посмотреть, например, в [1].

Обучение рекуррентной сети. Пусть среднеквадратичная ошибка одного входного объекта

$E(t) = \frac{1}{2}(y(t) - e)^2$, где e — эталонное значение для соответствующего входного объекта. Для определения алгоритма обратного распространения ошибки найдем производные функции среднеквадратичной ошибки по настраиваемым параметрам сети. Тогда

$$\frac{\partial E}{\partial v_i} = \frac{\partial E}{\partial y} \frac{\partial y}{\partial v_i} = (y(t) - e) p_i(t);$$

$$\frac{\partial E}{\partial T} = \frac{\partial E}{\partial y} \frac{\partial y}{\partial T} = -(y(t) - e);$$

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial y} \frac{\partial y}{\partial p_i} \frac{\partial p_i}{\partial S_i} \frac{\partial S_i}{\partial w_{ij}} = (y(t) - e) v_i F'(S_i) x_j(t);$$

$$\frac{\partial E}{\partial u_i} = \frac{\partial E}{\partial y} \frac{\partial y}{\partial p_i} \frac{\partial p_i}{\partial S_i} \frac{\partial S_i}{\partial u_i} = (y(t) - e) v_i F'(S_i) y(t-1);$$

$$\frac{\partial E}{\partial T_i} = \frac{\partial E}{\partial y} \frac{\partial y}{\partial p_i} \frac{\partial p_i}{\partial S_i} \frac{\partial S_i}{\partial T_i} = -(y(t) - e) v_i F'(S_i),$$

где $F'(S_i)$ — производная функции активации промежуточного слоя

Недостатки алгоритма обратного распространения ошибки. Алгоритм обратного распространения ошибки, в основе которого лежит градиентный метод, создает ряд проблем при обучении многослойных нейронных сетей. Существуют следующие проблемы:

- неизвестность выбора числа слоев и количества нейронных элементов в слое для многослойной сети,
- медленная сходимость градиентного метода с постоянным шагом обучения,
- сложность выбора подходящей скорости обучения,
- невозможность отделения точек локального и глобального минимума (градиентный метод их не различает).

Способ повышения скорости обучения. В алгоритме обратного распространения ошибки чем больше параметр обучения a , тем больше корректировка весов и, следовательно, тем выше скорость обучения. Однако, в этом случае система может перейти в неустойчивое состояние. Простейшим способом повышения скорости обучения без потери устойчивости является добавление зависимости текущего изменения весов от изменений весов на предыдущих шагах алгоритма с какими-то коэффициентами η (так называемое обобщенное дельта-правило).

$$w_{ij}(t+1) = w_{ij}(t) + \eta \Delta w_{ij}(t-1) - a \frac{\partial E(t)}{\partial w_{ij}(t)},$$

где

$$\Delta w_{ij}(t-1) = \eta \Delta w_{ij}(t-2) - a \frac{\partial E(t-1)}{\partial w_{ij}(t-1)}.$$

Особенности обобщенного дельта-правила:

1. Текущее значение коррекции весов $\Delta w_{ij}(t)$ представляет собой сумму экспоненциально взвешенного ряда. Для того чтобы ряд сходиллся, постоянная η должна находиться в диапазоне $0 < |\eta| < 1$ (см. [7]). Она может быть и отрицательной, хотя это не рекомендуется на практике.
2. Если частная производная $\frac{\partial E(t)}{\partial w_{ij}(t)}$ имеет один и тот же знак на нескольких последовательных итерациях, то экспоненциально взвешенная сумма $\Delta w_{ij}(t)$ возрастает по абсолютному значению, поэтому веса могут меняться на значительную величину (ускорение спуска).
3. Если частная производная $\frac{\partial E(t)}{\partial w_{ij}(t)}$ на нескольких итерациях меняет знак, то экспоненциально взвешенная сумма $\Delta w_{ij}(t)$ уменьшается по абсолютной величине, поэтому веса $w_{ij}(t)$ меняются незначительно.

Вычислительный эксперимент

Данные. В качестве периодического, зашумленного и апериодического временных рядов используются модельные данные:

- $y = 0.4 \sin(5x) + 0.5$,
- $y = 0.2 \sin(20x) + 0.3 \cos(15x) + 0.1 \cos(5x) + 0.3 \sin(3x)$,
- $y = 0.1x + 0.1 \sin(x) + 0.05 \cos(30x)$.

Размерность обучающей выборки — L , число признаков у объекта — n , число нейронов в промежуточном слое — p , функция активации — F , шаг обучения — a , минимум ошибки — E_{min} , коэффициент в обобщенном дельта-правиле — η .

Исследуются две функции активации (Рис. 2):

1. $F(S) = \frac{1}{1+e^{-S}}$ (сигмоидная),
2. $F(S) = \frac{1-e^{-2S}}{1+e^{-2S}}$ (гиперболический тангенс).

Результаты.

1. $y = 0.4 \sin(5x) + 0.5$ (Рис. 3)

Минимум ошибки $E_{min} = 0.0001$, шаг обучения $a = 0.03$. Размерность обучающей выборки $L = 40$ со значениями, равномерно расположенными на отрезке $[0, 2\pi]$. Результаты: Таб. 1. Проиллюстрируем данные в таблице на графиках зависимости числа итераций от числа нейронов в промежуточном слое для каждой ширины окна (рис. 4). На графиках 4, 6 наблюдаем уменьшение числа итераций при увеличении числа нейронов промежуточного слоя. Результаты работы алгоритма при $n = 15$, $p = 15$, $F(S) = \frac{1-e^{-2S}}{1+e^{-2S}}$ и числе выходных нейронов 5 — Рис. 7.

2. $y = 0.2 \sin(20x) + 0.3 \cos(15x) + 0.1 \cos(5x) + 0.3 \sin(3x)$ (Рис. 8).

Минимум ошибки $E_{min} = 0.0001$, шаг обучения $a = 0.03$. Размерность обучающей выборки $L = 40$ со значениями, равномерно расположенными на отрезке $[0, 2\pi]$. Результаты: Таб. 2 Проиллюстрируем данные в таблице на графиках зависимости числа итераций от числа нейронов в промежуточном слое для каждой ширины окна (рис. 9).

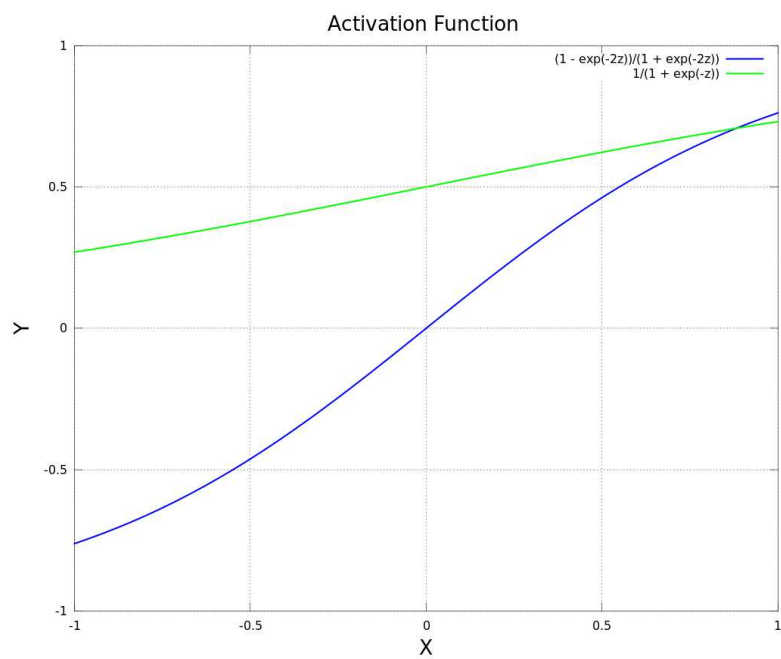


Рис. 2. Функции активации

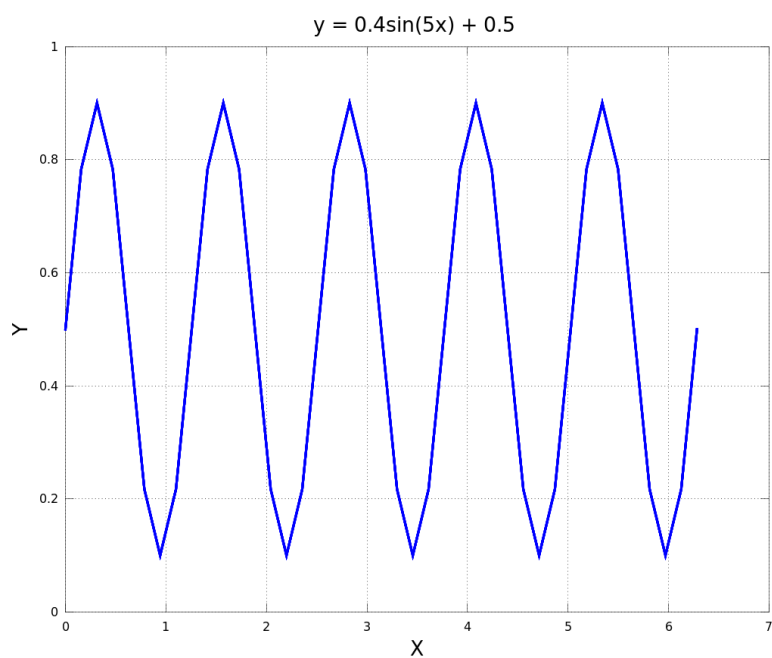


Рис. 3. График функции $y = 0.4\sin(5x) + 0.5$

Таблица 1. Число итераций и ошибка на контрольной выборке при прогнозировании временного ряда $y = 0.4 \sin(5x) + 0.5$ в зависимости от параметров

		Число нейронов промежуточного слоя p				
		3	6	9	12	15
Ширина окна n	3	146(0.0001) 44(0.002)	81(0.031) 19(0.007)	83(0.014) 27(0.0001)	95(0.013) 21(0.002)	76(0.021) 17(0.002)
	6	626(0.000012) 31(0.002)	159(0.00001) 28(0.0009)	120(0.0001) 24(0.0005)	85(0.007) 27(0.0004)	79(0.007) 26(0.0005)
	9	53(0.05) 33(0.0002)	82(0.01) 20(0.0009)	69(0.02) 22(0.0012)	66(0.016) 23(0.0001)	64(0.017) 21(0.0002)
	12	54(0.05) 18(0.004)	67(0.02) 21(0.0024)	68(0.02) 22(0.002)	62(0.02) 22(0.0024)	71(0.02) 21(0.0024)
	15	42(0.06) 9(0.003)	81(0.03) 11(0.001)	72(0.03) 18(0.006)	67(0.02) 7(0.0041)	67(0.02) 8(0.003)
	21	65(0.03) 21(0.003)	105(0.03) 17(0.002)	88(0.03) 16(0.002)	89(0.02) 18(0.001)	78(0.02) 16(0.005)

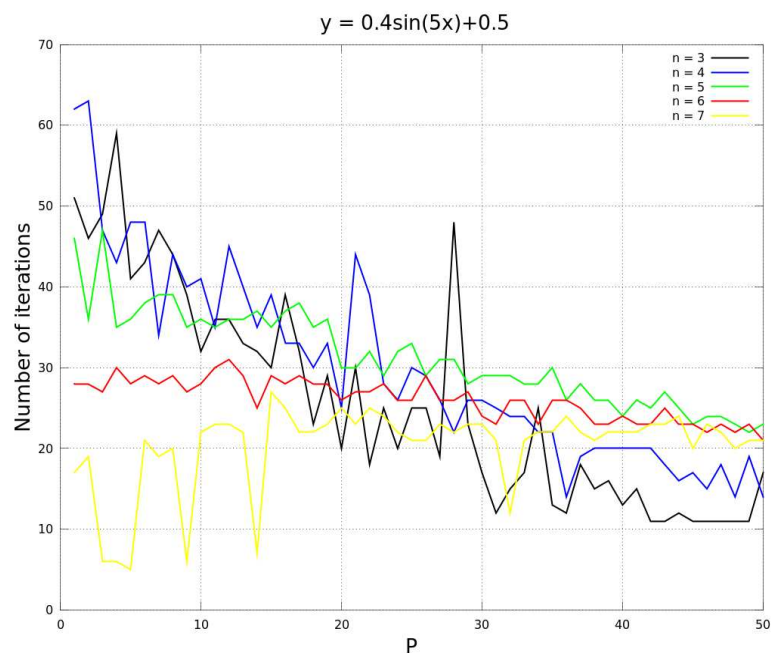


Рис. 4. Результаты для функции $y = 0.4 \sin(5x) + 0.5$ (ширина окна 3 - 7).

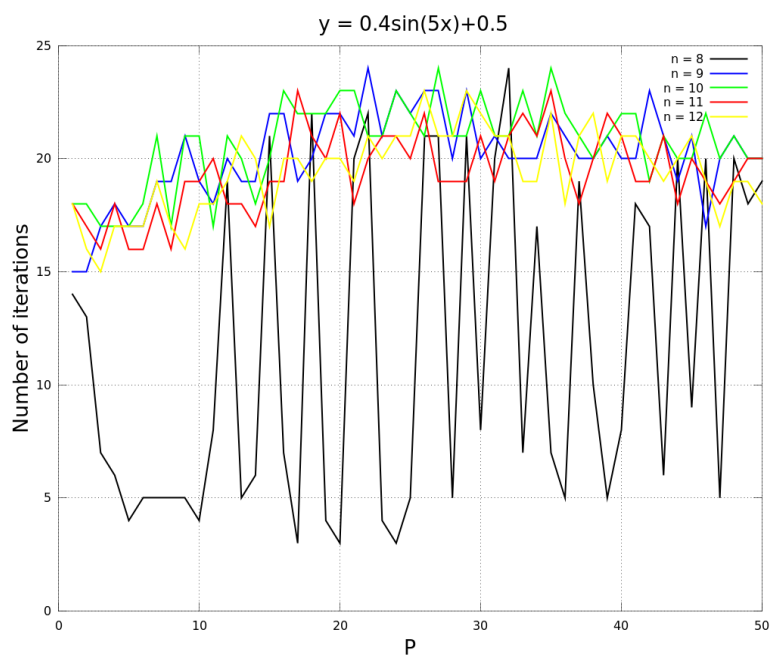


Рис. 5. Результаты для функции $y = 0.4\sin(5x) + 0.5$ (ширина окна 8 - 12).

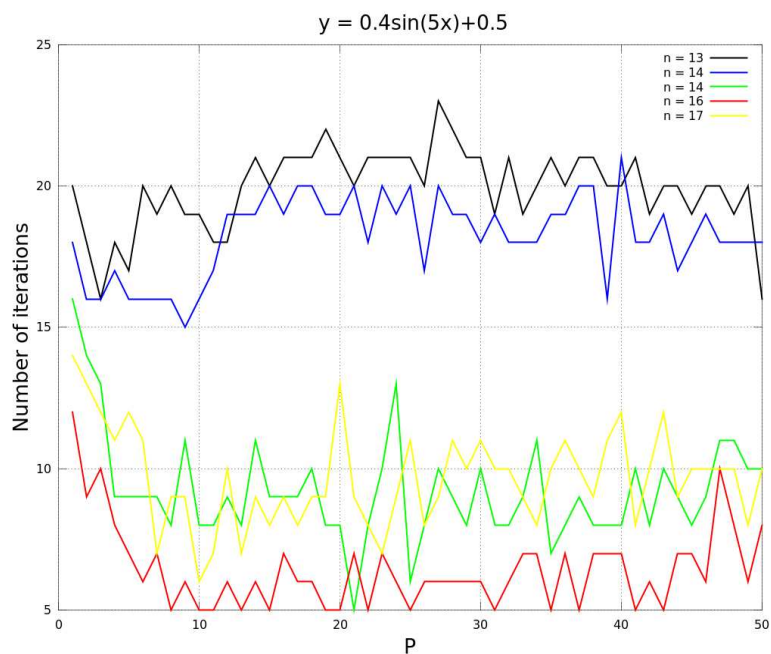


Рис. 6. Результаты для функции $y = 0.4\sin(5x) + 0.5$ (ширина окна 13 - 17).

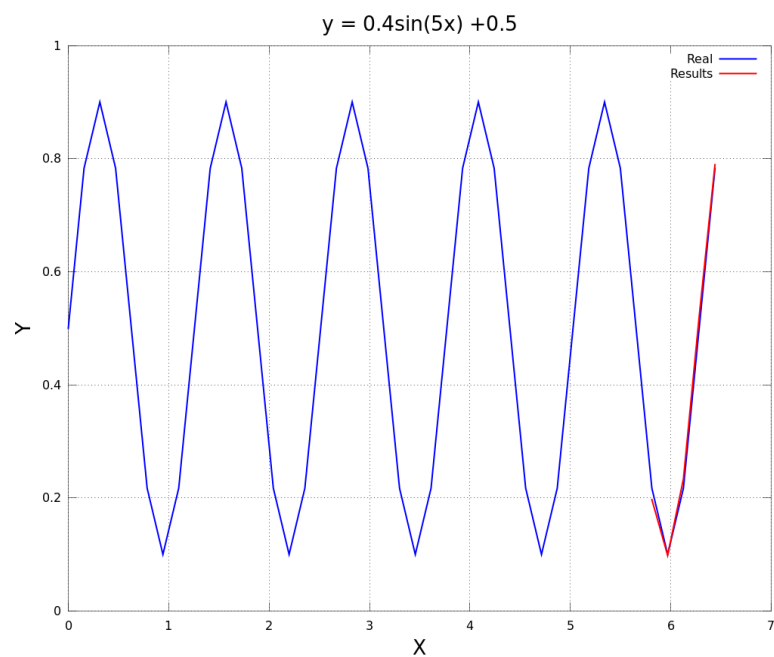


Рис. 7. Результаты для функции $y = 0.4\sin(5x) + 0.5$.

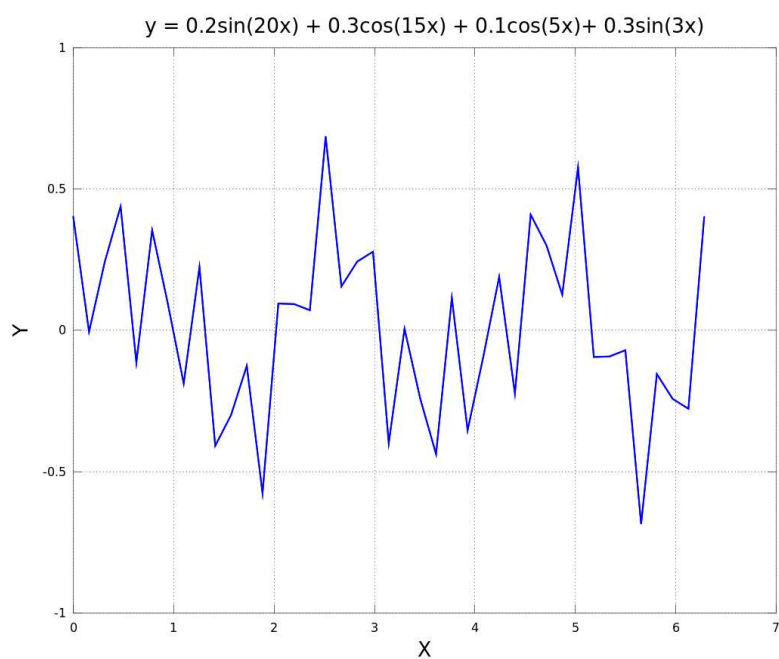


Рис. 8. График функции $y = 0.2\sin(20x) + 0.3\cos(15x) + 0.1\cos(5x) + 0.3\sin(3x)$.

Таблица 2. Число итераций и ошибка на контрольной выборке при прогнозировании временного ряда $y = 0.2 \sin(20x) + 0.3 \cos(15x) + 0.1 \cos(5x) + 0.3 \sin(3x)$ в зависимости от параметров.

		Число нейронов промежуточного слоя p				
		3	6	9	12	15
Ширина окна n	3	»2000 »2000	»2000 »2000	»2000 »2000	»2000 »2000	»2000 »2000
	6	426(0.006) 98(0.006)	358(0.007) 118(0.005)	322(0.008) 102(0.005)	225(0.01) 127(0.006)	181(0.017) 124(0.005)
	9	665(0.0003) 190(0.0003)	687(0.0002) 191(0.0001)	666(0.0002) 186(0.0001)	773(0.00009) 187(0.0001)	781(0.00005) 185(0.0002)
	12	427(0.0005) 99(0.00026)	411(0.0003) 117(0.00059)	390(0.0004) 105(0.0002)	405(0.0004) 106(0.0002)	421(0.0004) 93(0.00001)
	15	559(0.00006) 130(0.00005)	537(0.00006) 142(0.00005)	540(0.00003) 144(0.00005)	560 (0.00002) 170(0.00006)	534(0.00001) 151(0.00005)
	21	644(0.00004) 119(0.0001)	510(0.00005) 113(0.00005)	492(0.00003) 115(0.00004)	490 (0.000006) 117(0.0002)	517(0.0000001) 124(0.0003)

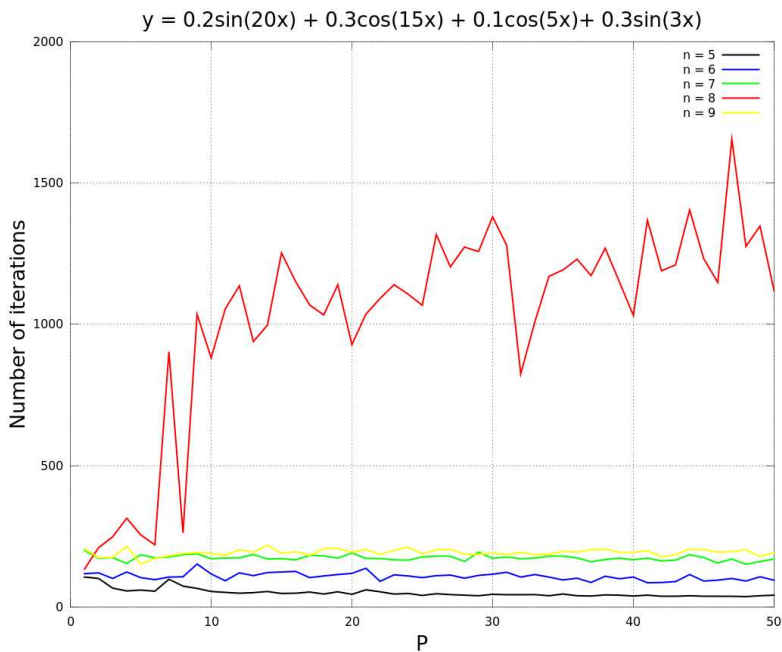


Рис. 9. Результаты для функции $y = 0.2 \sin(20x) + 0.3 \cos(15x) + 0.1 \cos(5x) + 0.3 \sin(3x)$ (ширина окна 5 - 9).

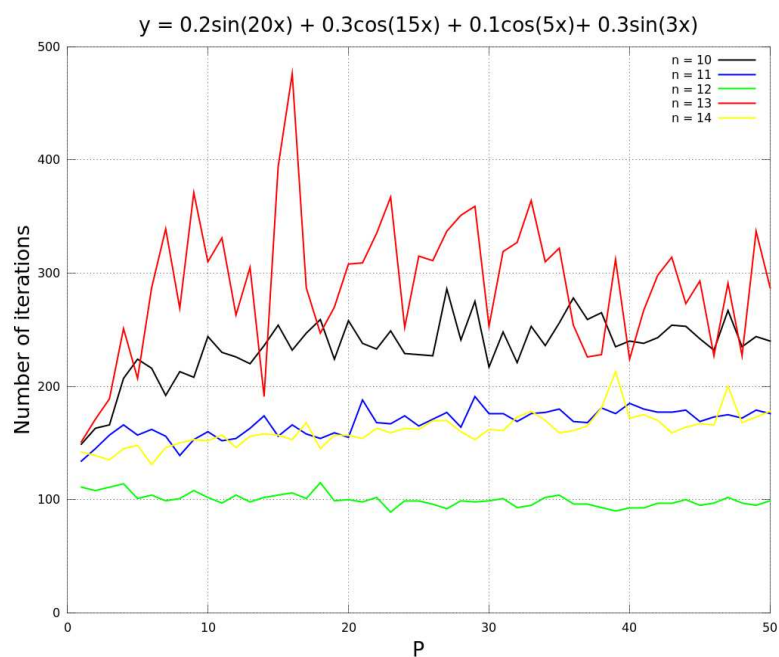


Рис. 10. Результаты для функции $y = 0.2 \sin(20x) + 0.3 \cos(15x) + 0.1 \cos(5x) + 0.3 \sin(3x)$ (ширина окна 9 - 14).

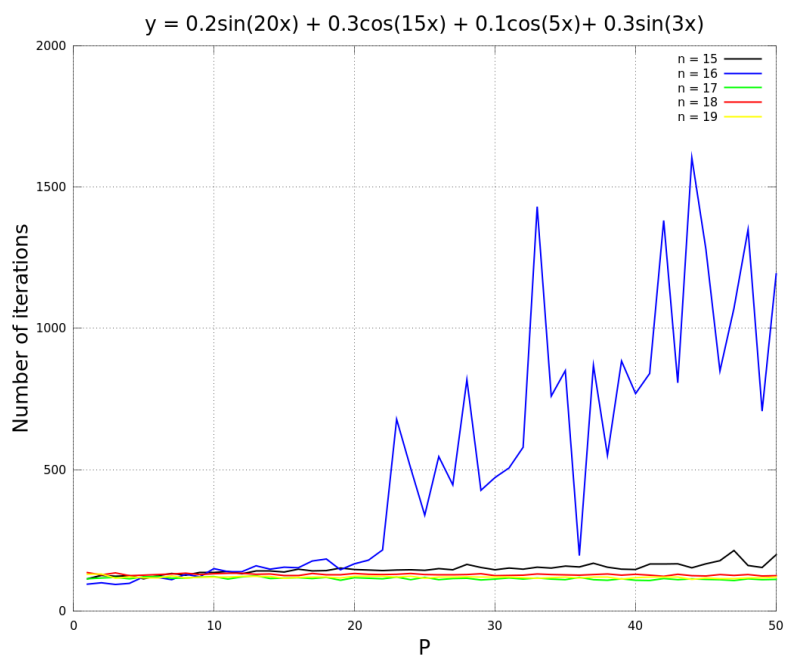


Рис. 11. Результаты для функции $y = 0.2 \sin(20x) + 0.3 \cos(15x) + 0.1 \cos(5x) + 0.3 \sin(3x)$ (ширина окна 15 - 19).

Не удалось выявить зависимости числа итераций как от числа нейронов промежуточного слоя, так и от ширины окна. Результаты работы алгоритма при $n = 12$, $p = 15$, $F(S) = \frac{1-e^{-2S}}{1+e^{-2S}}$ и числе выходных нейронов 5 — Рис. 12.

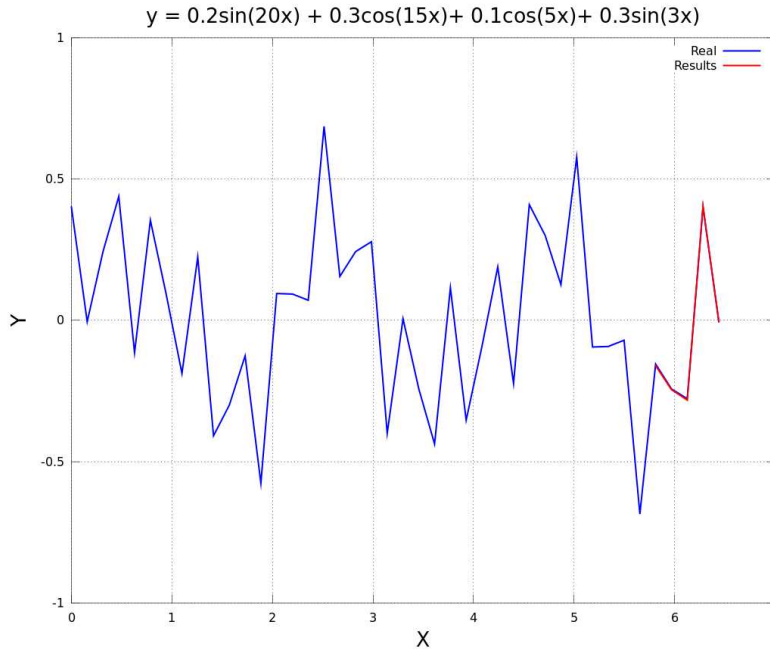


Рис. 12. Результаты алгоритма для функции $y = 0.2\sin(20x) + 0.3\cos(15x) + 0.1\cos(5x) + 0.3\sin(3x)$.

3. $y = 0.1x + 0.1\sin(x) + 0.05\cos(30x)$ (Рис. 13).

Минимум ошибки $E_{min} = 0.0001$, шаг обучения $a = 0.03$. Размерность обучающей выборки $L = 40$ со значениями, равномерно расположенными на отрезке $[0, 2\pi]$. Результаты: Таб. 3 Проиллюстрируем данные в таблице на графиках зависимости числа итераций от числа нейронов в промежуточном слое для каждой ширины окна (рис. 14).

На графиках 14,15,16 видно, что при увеличении числа нейронов в промежуточном слое число итераций сначала убывало, затем возрастало. Результаты алгоритма при $n = 15$, $p = 6$, $F(S) = \frac{1-e^{-2S}}{1+e^{-2S}}$ и числе выходных нейронов 5 — Рис. 17.

Для лучших (в смысле скорости сходимости) параметров каждой функции применим обобщенное дельта-правило с параметром $\eta = 0.2$ и сравним с первыми результатами (Таб. 4).

Заключение

В работе рассмотрена классическая архитектура рекуррентной нейронной сети с обратной связью (сеть Джордана), проанализированы на модельных данных зависимости скорости сходимости алгоритма обратного распространения ошибки от функций активации, числа нейронов промежуточного слоя и ширины окна, исследовано обобщенное дельта-правило. Получены результаты:

- использование в качестве функции активации тангенциальной функции повышает скорость сходимости,
- большая размерность промежуточного слоя повышает скорость сходимости,

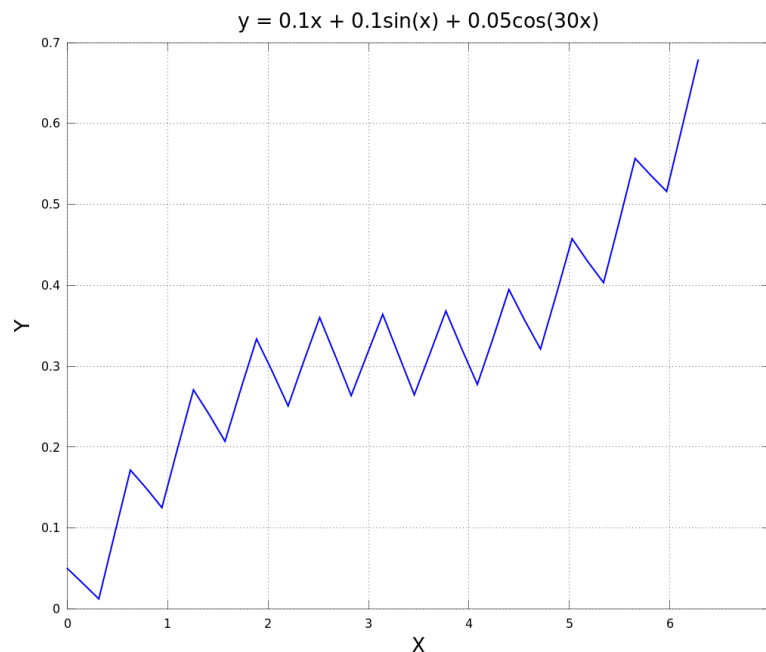


Рис. 13. График функции $y = 0.1x + 0.1\sin(x) + 0.05\cos(30x)$.

Таблица 3. Число итераций и ошибка на контрольной выборке при прогнозировании временного ряда $y = 0.1x + 0.1\sin(x) + 0.05\cos(30x)$ в зависимости от параметров

		Число нейронов промежуточного слоя p				
		3	6	9	12	15
Ширина окна n	3	»2000 »2000	»2000 »2000	»2000 »2000	»2000 »2000	»2000 »2000
	6	»2000 687(0.00005)	»2000 703(0.0002)	»2000 636(0.0001)	»2000 786(0.00002)	»2000 653(0.0002)
	8	»2000 720(0.00005)	»2000 827(0.0002)	»2000 811(0.0001)	»2000 825(0.00002)	»2000 831(0.0002)
	9	»2000 675(0.0003)	»2000 626(0.0001)	»2000 685(0.0002)	»2000 744(0.0004)	»2000 793(0.0002)
	12	2591(0.0001) 392(0.00086)	2432(0.0002) 412(0.00069)	2260(0.0004) 424(0.0002)	2134(0.0001) 489(0.0008)	2128(0.0005) 471(0.0006)
	15	2337(0.0001) 343(0.0004)	1994(0.00004) 325(0.0006)	2052(0.00003) 334(0.0005)	2026(0.00003) 358(0.0006)	2150(0.00005) 382(0.0006)
	21	4057(0.00001) 539(0.0004)	2929(0.00003) 501(0.0001)	2972(0.00003) 548(0.0001)	2972(0.00005) 529(0.00015)	2999(0.00005) 567(0.0002)

Таблица 4. Число итераций при использовании обобщенного дельта-правила в обучении сети.

	первые результаты	обобщенное дельта-правило
$y = 0.4\sin(5x) + 0.5$	8(0.003)	6(0.04)
$y = 0.2\sin(20x) + 0.3\cos(15x) + 0.1\cos(5x) + 0.3\sin(3x)$	91(0.00001)	81(0.0004)
$y = 0.1x + 0.1\sin(x) + 0.05\cos(30x)$	325(0.0006)	276(0.0006)

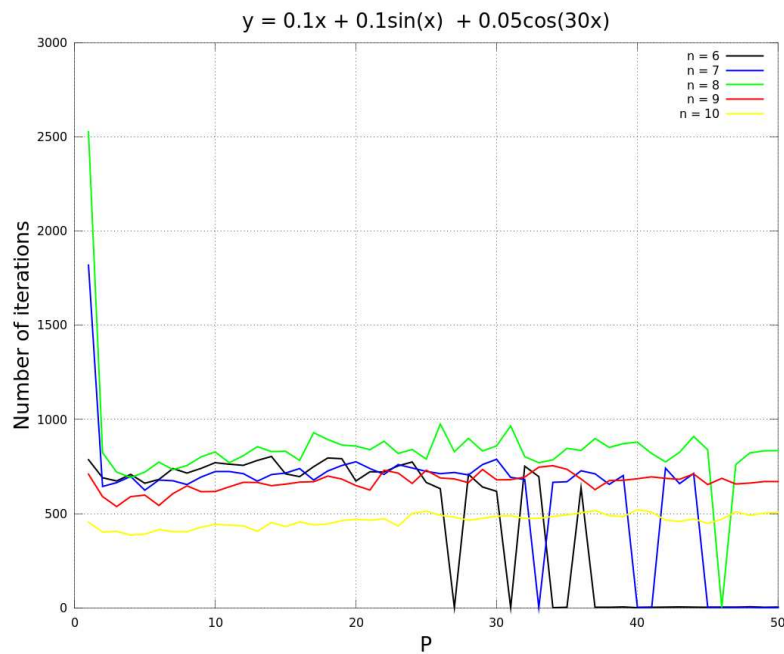


Рис. 14. Результаты для функции $y = 0.1x + 0.1\sin(x) + 0.05\cos(30x)$ (ширина окна 6 - 10).

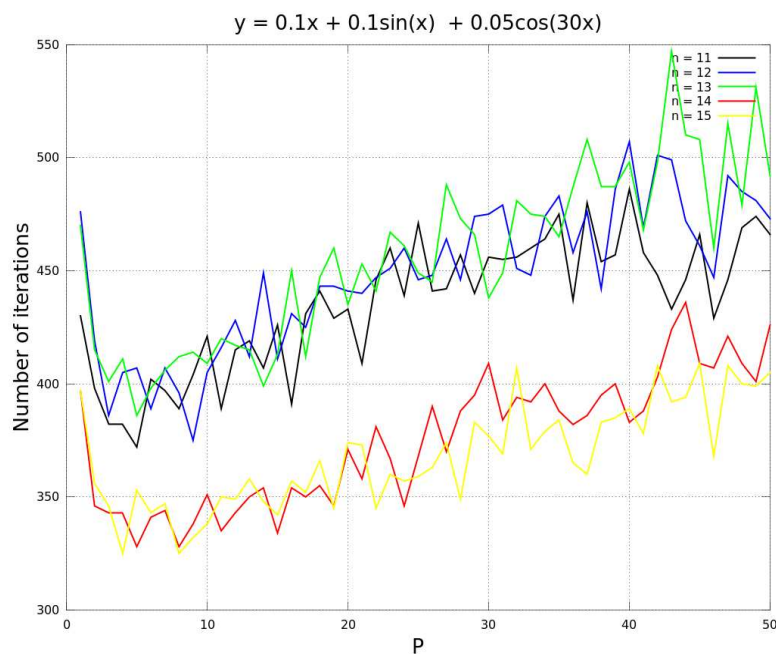


Рис. 15. Результаты для функции $y = 0.1x + 0.1\sin(x) + 0.05\cos(30x)$ (ширина окна 11 - 15).

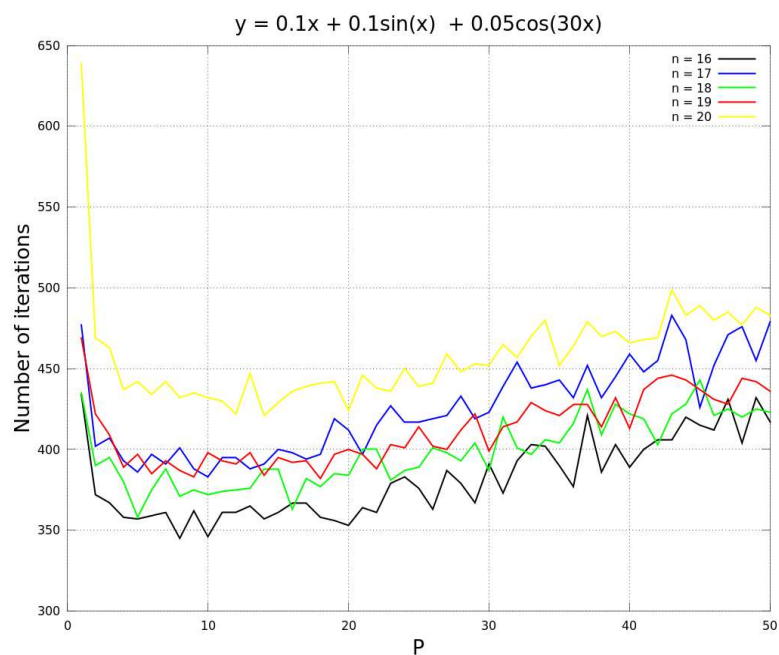


Рис. 16. Результаты для функции $y = 0.1x + 0.1\sin(x) + 0.05\cos(30x)$ (ширина окна 16 - 20).

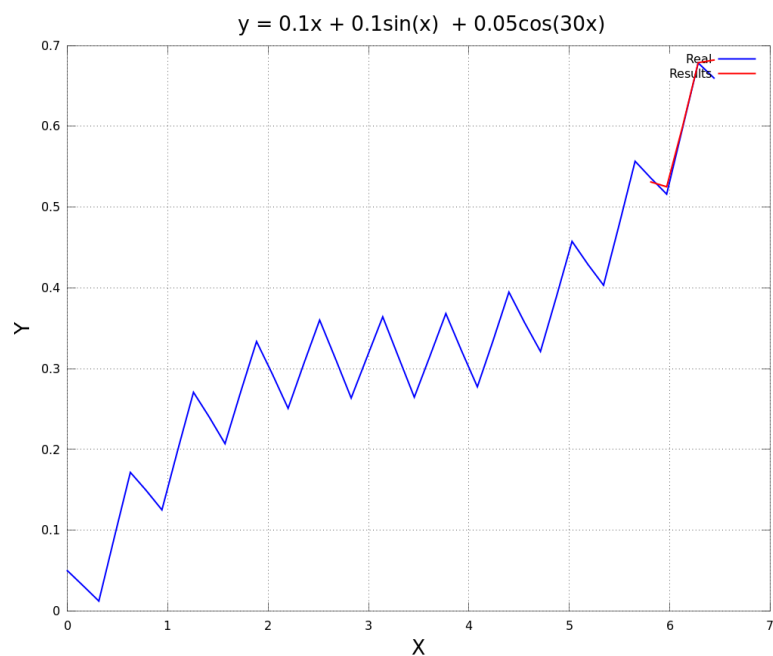


Рис. 17. Результаты алгоритма для функции $y = 0.1x + 0.1\sin(x) + 0.05\cos(30x)$ (8600 итераций).

- применение обобщенного дельта-правила повышает скорость сходимости,
- не найдено зависимости между скоростью сходимости и шириной окна,
- для каждого временного ряда существует некий оптимальный выбор, максимизирующий скорость сходимости.

Необходимый для повторения вычислительного эксперимента код можно найти на сайте:

<https://mlalgorithms.svn.sourceforge.net/svnroot/mlalgorithms/FNNForecasting/>

Литература

- [1] В. Головкин *Нейронные сети: обучение, организация и применение*, ИПРЖР, 2001.
- [2] David MacKay *Information Theory, Inference, and Learning Algorithms*, 2003.
- [3] W. S. McCulloch and W. Pitts *A Logical Calculus of Ideas Immanent in Nervous Activity*, Bulletin of Mathematical Biophysics, 1943, Vol.5, pp.115–133.
- [4] M. Minsky and S. Papert *Perceptrons: An Introduction to Computational Geometry*, MIT Press, 1969.
- [5] D. Rumelhart and G. Hinton and R. Williams *Learning representations by back-propagating errors*, Nature, 1986, Vol.323, pp.533–536.
- [6] К. Воронцов *Лекции по искусственным нейронным сетям*, 2009, р.20.
- [7] С. Хайкин *Нейронные сети*, Вильямс, 2006.