

# ОБ ОДНОЙ ЗАДАЧЕ УПРАВЛЕНИЯ CALL-ЦЕНТРОМ С ПРИМЕНЕНИЕМ НЕЙРОННЫХ СЕТЕЙ

Томашевич Н.С., Томашевич Д.С., Галушкин А.И., Муромский М.Ю., Шачнев Е.А.

Научный центр нейрокомпьютеров  
107066, Москва, ул. Новая Басманная, д. 20

Проведено исследование различных структур нейронных сетей для решения задачи экстраполяции числа звонков Call-центра с учетом и без учета артефактов. Рассмотрены следующие структуры нейронных сетей: фиксированная структура с полными последовательными связями, фиксированная структура с обратными связями, переменная структура. Результаты экспериментов показали превосходство переменной структуры для решения данной задачи.

## Введение

Call-центры – это организации, занимающиеся профессиональной обработкой телефонных звонков. Непосредственной работой с клиентами Call-центра занимаются агенты. Каждый агент имеет какую-либо квалификацию или несколько квалификаций. Под квалификацией понимается умение агента принимать заказы на пиццу, авиабилеты и т.д. Обучение агента и получение им квалификации происходит за счет Call-центра и стоит достаточно дорого. Сама рабочая сила агентов также стоит дорого.

Одной из важных задач, стоящей перед Call-центром является правильное распределение рабочей силы таким образом, чтобы ни один агент не простаивал без работы, и при этом ни один звонок клиента не был потерян. Данная оптимизационная задача включает в себя следующую не менее сложную подзадачу: требуется сформировать прогноз числа звонков на месяц вперед для того, чтобы менеджер Call-центра мог правильно распределить рабочую силу.

В данной работе приведено описание различных методик построения нейронных сетей различной структуры для решения задачи экстраполяции числа звонков Call-центра. При решении данной задачи авторами использовалась база данных одного Call-центра, содержащая данные о числе звонков в течение года с разбивкой по полчаса, а также данные об артефактах, то есть о различных событиях, могущих как-то повлиять на число звонков.

## Нейросетевая постановка задачи

Число звонков из базы данных Call-центра представляет собой временной ряд, заданный на интервале  $[1, M]$ . Известно, что на поведение данного временного ряда могут оказывать влияние  $A$  различных видов артефактов, при этом в каждый момент времени  $n \in \{1, \dots, M\}$  может происходить любое число артефактов в любой комбинации.

Поскольку мы решаем задачу экстраполяции временного ряда, то будем формировать обучающую выборку посредством наблюдения интервала ряда длиной  $T$  и упреждения  $\alpha$  следующих значений ряда. То есть, в каждый момент времени  $n \in \{T, \dots, M - \alpha\}$  формируется элемент обучающей выборки, в которой **входным сигналом** нейронной сети являются:

вектор значений функции  $F$  (размерностью  $T$ ) на интервале наблюдения  $[n - T + 1, n]$ ;

$(T + \alpha)$  векторов артефактов (каждый размерностью  $A$ ), состоящих из 0 и 1. Если на интервале  $[n - T + 1, n + \alpha]$  в  $j$ -ый момент времени,  $j \in \{1, \dots, T + \alpha\}$ , встречается  $i$ -ый вид артефакта,  $i \in \{1, \dots, A\}$ , то в  $j$ -м векторе  $i$ -ый элемент устанавливается равным 1, в противном случае он равен 0.

И **желаемым выходным сигналом** является вектор значений функции  $F$  на интервале  $[n + 1, n + \alpha]$  размерностью  $\alpha$ . Таким образом, число обучающих примеров равно  $M - T - \alpha$ .

**Выходным сигналом** нейронной сети в момент времени  $n \in \{T, \dots, M - \alpha\}$  является вектор (размерностью  $\alpha$ ) упрежденных на интервале  $[n + 1, n + \alpha]$  значений функции  $F$ , обозначаемый через  $\bar{y}(n) = \{y_1(n), \dots, y_\alpha(n)\}$ .

**Ошибка решения** определяется как разность между желаемым и действительным выходным сигналом нейронной сети в дискретный момент времени  $n \in \{T, \dots, M - \alpha\}$ , то есть ошибка решения показывает расстояние до правильного значения. В нашем случае, ошибка решения в момент времени  $n$  представляет собой вектор,  $j$ -ый элемент которого  $e_j(n)$  равен:

$$e_j(n) = y_j(n) - F(n + j), \quad (2)$$

где  $n$  - текущий момент времени,  $n \in \{T, \dots, M - \alpha\}$ ;

$j$  - текущая позиция на интервале упреждения,  $j \in \{1, \dots, \alpha\}$ .

**Функция активации** должна быть монотонной с вещественным выходным сигналом. В решаемой в данной работе задаче функция  $F(x)$  принимает неотрицательные значения. Диапазон изменения значений функции составляет  $[0, 2m]$ , где  $2m$  - максимальное значение функции  $F$  на интервале  $[1, M]$ . Поэтому, будем использовать функцию активации типа арктангенса вида:

$$y_{h_k}(n) = m \left( \frac{2}{\pi} \arctg(g_{h_k}(n)) \right) + m, \quad (3)$$

где  $y_{h_k}(n)$  - выход  $h_k$ -го нейрона  $k$ -го слоя сети,  $k \in \{1, \dots, K\}$ , в момент времени  $n$ ;

$g_{h_k}(n)$  - активация  $h_k$ -го нейрона  $k$ -го слоя сети,  $k \in \{1, \dots, K\}$ , в момент времени  $n$ .

**Функционал оптимизации**  $\Phi(n)$  использовался один для всех структур нейронных сетей, а именно, минимизирующий среднеквадратичную ошибку. При обучении на  $n$ -ом примере,  $n \in \{T, \dots, M - \alpha\}$ , функционал выглядит следующим образом:

$$\Phi(n) = \min \frac{1}{2} \sum_{h_k=1}^{\alpha} e_{h_k}^2(n), \quad (4)$$

где  $e_{h_k}(n)$  - ошибка  $h_k$ -го нейрона выходного слоя, вычисляющаяся по формуле (2).

Поиск экстремума (минимума) функционала оптимизации для всех структур нейронных сетей проводился **градиентным методом со случайным выбором начальных условий**. Его суть заключается в том, что начальные значения весов сети выбираются случайным образом, и далее при обучении сети после  $n$ -ой итерации,  $n \in \{T, \dots, M - \alpha\}$ , каждой эпохи вычисляется ошибка, и все веса сети обновляются следующим образом:

$$w_{h_k, h_{(k-1)}}(n+1) = w_{h_k, h_{(k-1)}}(n) - K_k^* \frac{d\Phi(n)}{dw_{h_k, h_{(k-1)}}}, \quad (5)$$

где  $K_k^*$  - это коэффициент скорости обучения  $k$ -го слоя сети, определяющий ширину шага по поверхности ошибки.

Построение многослойной нейронной сети фиксированной структуры с полными последовательными связями

Структура разомкнутой  $K$ -слойной сети с полными последовательными связями представлена на рис. 1 и описывается уравнением (3) и следующим уравнением:

$$g_{h_k}(n) = \sum_{h_{(k-1)}=0}^{H_{(k-1)}} w_{h_k, h_{(k-1)}} \cdot y_{h_{(k-1)}}(n), \quad (6)$$

где  $H_{(k-1)}$  - количество нейронов  $(k-1)$ -го слоя; при  $(k-1) = 0$ ,  $H_0 = T$ ;

$y_{h_{(k-1)}}(n)$  - выходной сигнал  $h_{(k-1)}$ -го нейрона  $(k-1)$ -го слоя НС; при  $h_{(k-1)} = 0$ ,  $y_0(n) = 1$ ; при  $(k-1) = 0$  и  $h_0 > 0$ ,  $y_{h_0}(n) = F(n - h_0 + 1)$ ;

$w_{h_k, h_{(k-1)}}$  - вес связи от  $h_{(k-1)}$ -го нейрона  $(k-1)$ -го слоя к  $h_k$ -му нейрону  $k$ -го слоя.

Число настраиваемых параметров  $K$ -слойной нейронной сети равно:

$$\sum_{k=1}^K (H_{(k-1)} + 1) \cdot H_k, \quad (7)$$

где  $H_{(k-1)}$  - количество нейронов  $(k-1)$ -го слоя; при  $(k-1) = 0$ ,  $H_0 = T$ ;

$H_k$  - количество нейронов  $k$ -го слоя НС.

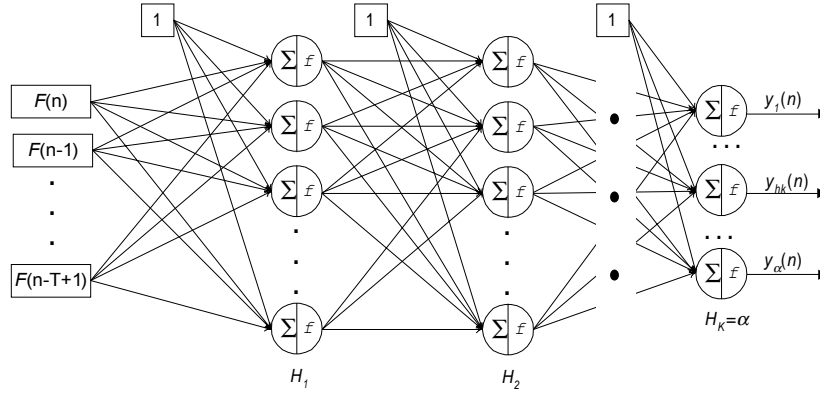


Рис. 1. Структура разомкнутой  $K$ -слойной сети с полными последовательными связями

Построение трехслойной нейронной сети фиксированной структуры с полными последовательными и обратными связями

Структура трехслойной нейронной сети с полными последовательными и обратными связями представлена на рис. 2 и описывается уравнением (3) и следующими уравнениями:

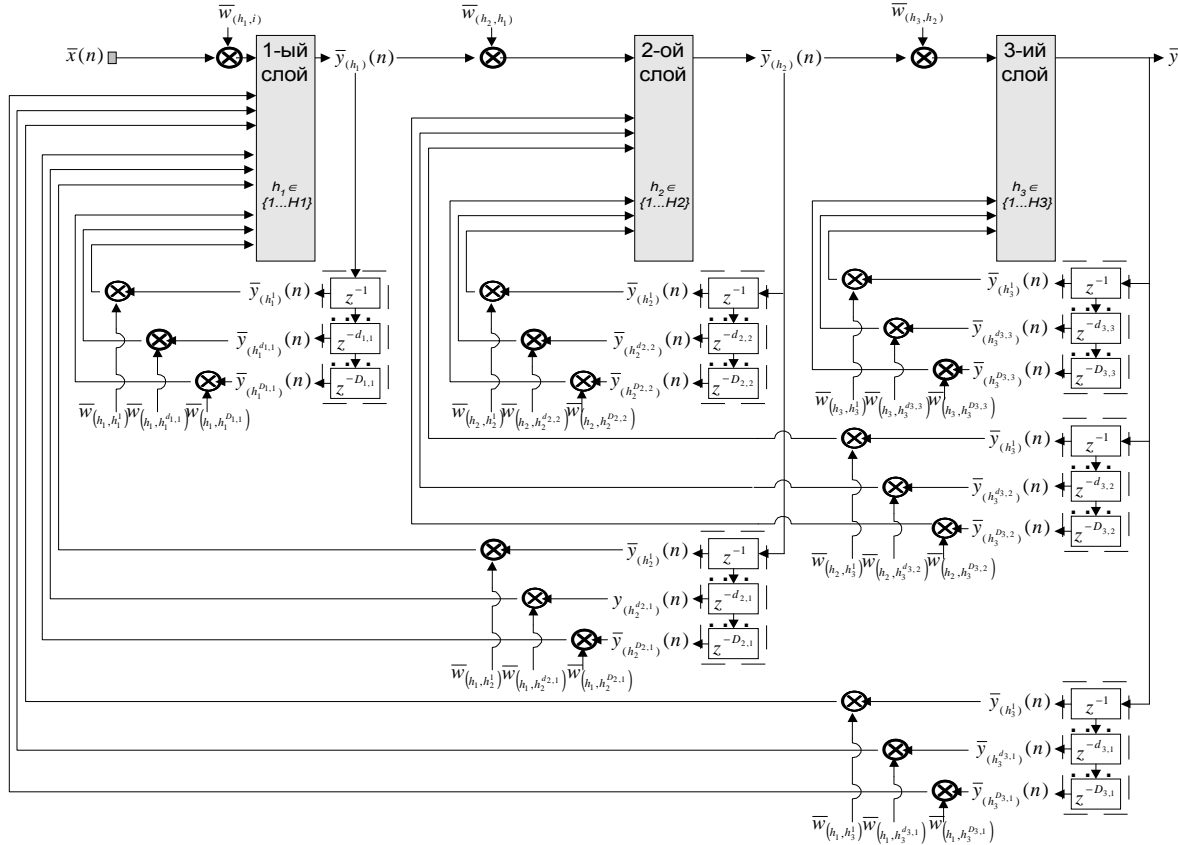


Рис. 2. Структура разомкнутой трехслойной сети с полными последовательными и обратными связями

Активация  $h_1$ -го нейрона 1-го слоя сети  $g_{h_1}(n)$  в момент времени  $n \in \{T, \dots, M - \alpha\}$  вычисляется по формуле:

$$g_{h_1}(n) = \left( w_{h_1,0} + \sum_{i=1}^T w_{h_1,i} \cdot F(n-i+1) \right) + \sum_{k=1}^3 \sum_{d_{k,1}=1}^{D_{k,1}} \sum_{h_k^{d_{k,1}}=1}^{H_k} w_{h_1, h_k^{d_{k,1}}} \cdot y_{h_k^{d_{k,1}}}(n), \quad (8)$$

где  $k$  - индекс по номерам слоев,  $k \in \{1, 2, 3\}$ ;

$d_{k,1}$  - номер блока задержки, который выполняет задержку выходных значений  $k$ -го слоя перед их подачей на вход 1-го слоя;

$D_{k,1}$  - размер банка памяти (или максимальное количество блоков задержки), при подаче выходных значений  $k$ -го,  $k = \{1,2,3\}$ , слоя на вход 1-го слоя НС;

$H_k$  - число нейронов  $k$ -го слоя сети,  $k = \{1,2,3\}$ ;

$h_k^{d_{k,1}}$  -  $h_k$ -ый элемент  $d_{k,1}$ -го блока задержки,  $k = \{1,2,3\}$ ;

$w_{h_1, h_k^{d_{k,1}}}$  - вес от  $h_k$ -го элемента  $d_{k,1}$ -го блока задержки,  $k = \{1,2,3\}$ , к  $h_1$ -му нейрону 1-го слоя сети,  $h_1 \in \{1, \dots, H_1\}$ ;

$w_{h_1, i}$  - вес от  $i$ -го входа к  $h_1$ -му нейрону 1-го слоя НС;

$y_{h_k^{d_{k,1}}}(n)$  - выход  $h_k$ -го элемента  $d_{k,1}$ -го блока задержки,  $k = \{1,2,3\}$ , в момент времени  $n$ .

Активация  $h_2$ -го нейрона 2-го слоя сети  $g_{h_2}(n)$  в момент времени  $n \in \{T, \dots, M - \alpha\}$  вычисляется по формуле:

$$g_{h_2}(n) = \sum_{h_1=0}^{H_1} w_{h_2, h_1} \cdot y_{h_1}(n) + \sum_{k=2}^3 \sum_{d_{k,2}=1}^{D_{k,2}} \sum_{h_k^{d_{k,2}}=1}^{H_k} w_{h_2, h_k^{d_{k,2}}} \cdot y_{h_k^{d_{k,2}}}(n), \quad (9)$$

где  $d_{k,2}$  - номер блока задержки, который выполняет задержку выходных значений  $k$ -го слоя перед их подачей на вход 2-го слоя;

$D_{k,2}$  - размер банка памяти (или максимальное количество блоков задержки), при подаче выходных значений  $k$ -го,  $k = \{2,3\}$ , слоя на вход 2-го слоя НС;

$h_k^{d_{k,2}}$  -  $h_k$ -ый элемент  $d_{k,2}$ -го блока задержки,  $k = \{2,3\}$ ;

$w_{h_2, h_k^{d_{k,2}}}$  - вес от  $h_k$ -го элемента  $d_{k,2}$ -го блока задержки,  $k = \{2,3\}$ , к  $h_2$ -му нейрону 2-го слоя сети,  $h_2 \in \{1, \dots, H_2\}$ ;

$w_{h_2, h_1}$  - вес от  $h_1$ -го нейрона 1-го слоя НС к  $h_2$ -му нейрону 2-го слоя НС;

$y_{h_k^{d_{k,2}}}(n)$  - выход  $h_k$ -го элемента  $d_{k,2}$ -го блока задержки,  $k = \{2,3\}$ , в момент времени  $n$ .

Активация  $h_3$ -го нейрона 3-го слоя сети  $g_{h_3}(n)$  в момент времени  $n \in \{T, \dots, M - \alpha\}$  вычисляется по формуле:

$$g_{h_3}(n) = \sum_{h_2=0}^{H_2} w_{h_3, h_2} \cdot y_{h_2}(n) + \sum_{d_{3,3}=1}^{D_{3,3}} \sum_{h_3^{d_{3,3}}=1}^{H_3} w_{h_3, h_3^{d_{3,3}}} \cdot y_{h_3^{d_{3,3}}}(n), \quad (10)$$

где  $d_{3,3}$  - номер блока задержки, который выполняет задержку выходных значений 3-го слоя перед их подачей на вход 3-го слоя;

$D_{3,3}$  - размер банка памяти (или максимальное количество блоков задержки), при подаче выходных значений 3-го слоя на вход 3-го слоя НС;

$h_3^{d_{3,3}}$  -  $h_3$ -ий элемент  $d_{3,3}$ -го блока задержки;

$w_{h_3, h_3^{d_{3,3}}}$  - вес от  $h_3$ -го элемента  $d_{3,3}$ -го блока задержки к  $h_3$ -му нейрону 3-го слоя сети,  $h_3 \in \{1, \dots, H_3\}$ ;

$w_{h_3, h_2}$  - вес от  $h_2$ -го нейрона 2-го слоя НС к  $h_3$ -му нейрону 3-го слоя НС;

$y_{h_3^{d_{3,3}}}(n)$  - выход  $h_3$ -го элемента  $d_{3,3}$ -го блока задержки в момент времени  $n$ .

Построение многослойной нейронной сети переменной структуры

Структура сети называется «переменной», потому что число нейронов и число слоев сети заранее не известно, а определяется в процессе создания сети. После создания сеть имеет ту же структуру, что и сеть на рис. 1. Сколько слоев и нейронов в слоях будет в НС сказать заранее нельзя. Это определяется в процессе обучения НС и зависит от сложности аппроксимируемой функции  $F$ . Суть создания сети с переменной структурой заключается в следующем.

Сперва мы строим сеть, состоящую из одного нейрона в первом слое и  $\alpha$  нейронов во втором (выходном) слое. Эта структура представляет собой обучаемый блок, показанный на рис. 3. Для этого обучаемого блока делается какое-то, заранее заданное пользователем, количество случайных выбросов

начальных значений весов,  $N_{vibros}$ . После каждого выброса вычисляется ошибка сети. Когда произведены все выбросы, для обучаемого блока выбираются те значения весов, которые соответствуют наименьшей ошибке. После этого, обучаемый блок обучается на  $N_{epoch}$  эпохах.

Назовем описанный выше процесс **этапом обучения блока** (напомню, этап состоит из  $N_{vibros}$  случайных выбросов начальных значений весов и  $N_{epoch}$  обучающих эпох).

Затем, в первый слой добавляется еще один нейрон, а также добавляются связи от этого нейрона к нейронам второго слоя. Теперь обучаемый блок включает в себя второй нейрон первого слоя (со всеми входящими и выходящими из него связями), а также  $\alpha$  нейронов второго слоя, но только со связями от второго нейрона первого слоя и смещениями. Для этого обучаемого блока также выполняется этап обучения.

Аналогичным образом нейроны продолжают добавляться в первый слой. При этом, каждый нейрон должен улучшать общее качество аппроксимации ряда сетью. Качество аппроксимации ряда оценивается с помощью ошибки, усредненной по всем примерам и по всем выходам:

$$E = \frac{1}{M - \alpha - T} \frac{1}{\alpha} \sum_{n=T}^{M-\alpha} \sum_{h_k=1}^{\alpha} |e_{h_k}(n)|, \quad (11)$$

где  $e_{h_k}(n)$  - ошибка  $h_k$ -го нейрона выходного  $K$ -го слоя НС, полученная на  $n$ -м примере.

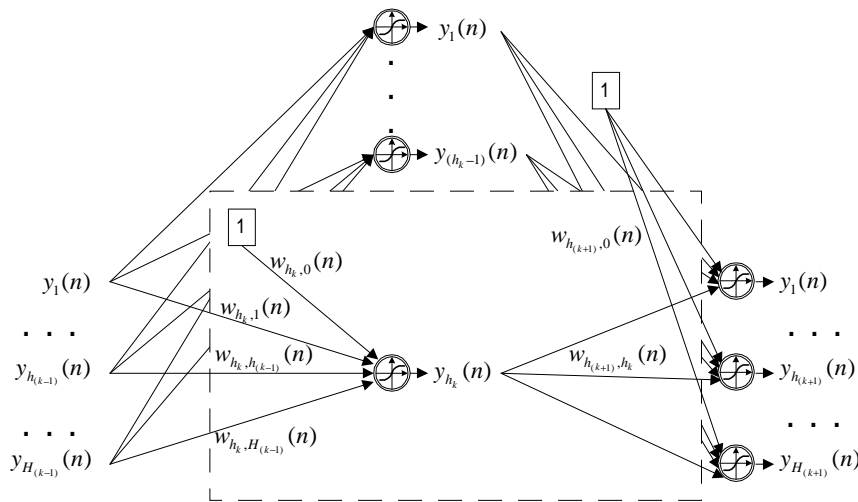


Рис. 3. Обучаемый блок  $h_k$ -го нейрона

Если очередной, добавленный в первый слой нейрон, не улучшает качество аппроксимации, то этап обучения повторяется. Обозначим максимальное число повторений этапа обучения для одного нейрона через  $N_{povtor}$ . Если через  $N_{povtor}$  повторений текущий нейрон не стал улучшать качество аппроксимации (т.е. ошибка текущего нейрона не стала меньше ошибки предыдущего нейрона), то текущий нейрон удаляется, а в сеть добавляется третий слой, состоящий из  $\alpha$  нейронов. Теперь случайным образом выбираются только веса между  $\alpha$  нейронами 2-го слоя и  $\alpha$  нейронами 3-го слоя. При этом также производится  $N_{vibros}$  случайных выбросов начальных значений этих весов, затем проводится  $N_{epoch}$  обучающих эпох, и если добавленный слой не стал улучшать качество аппроксимации, то вышеописанный процесс повторяется максимум  $N_{povtor}$  раз. Если через  $N_{povtor}$  повторений 3-й слой так и не стал улучшать качество аппроксимации, то он удаляется и считается, что при заданных значениях  $N_{vibros}$ ,  $N_{epoch}$  и  $N_{povtor}$  нельзя достичь лучшего качества аппроксимации. Если же 3-й слой улучшил качество аппроксимации, то во 2-й слой продолжают добавляться нейроны таким же образом, как и в 1-й слой.

Подобным образом создается столько слоев, сколько необходимо для достижения требуемого качества аппроксимации временного ряда.

Для уменьшения структуры сети и увеличения аппроксимирующих свойств каждого нейрона, полезно ввести параметр  $\Delta_{best}$ , показывающий на сколько каждый нейрон должен улучшать общее качество аппроксимации функции  $F$  нейронной сетью.

#### Результаты экспериментов и выводы

Были проведены эксперименты на описанных выше трех типах структур нейронных сетей. При этом для каждой структуры НС проводилось два типа экспериментов: с учетом артефактов в обучающей выборке, и без учета артефактов в обучающей выборке. Были получены следующие результаты.

Ошибка обучения на трехслойной НС фиксированной структуры с полными последовательными связями без учета артефактов составила 10% от максимального значения функции, тестовая ошибка – 5%. Ошибка обучения на той же структуре, но с учетом артефактов, составила 6%, а тестовая ошибка – 12%. Такое увеличение тестовой ошибки вызвано добавлением в сеть очень большого числа весов от артефактовых входов (всего в обучающей выборке было 20 артефактов, значение интервала наблюдения  $T$  было взято 30 дней, то есть число артефактовых весов равно 600). Учтывая, что на всем временном ряде число артефактов одного вида редко превышало 5, становится понятно, что веса от артефактовых входов были неспособны обучиться на таком малом числе примеров, и приводили к зашумлению выходного сигнала нейронной сети.

Эксперименты на трехслойной НС фиксированной структуры с полными последовательными и обратными связями проводились без учета артефактов, ошибки обучения и тестирования составили около 25% каждая. Такое резкое увеличение ошибок было вызвано введением большого числа весов, необходимых для реализации обратной связи с хотя бы единичной задержкой. Поскольку авторы не знали, как правильно выбирать начальные значения весов обратных связей, а выбор весов случайным образом приводил к очень большой ошибке, то эксперименты на данной структуре с учетом артефактов не проводились.

В результате экспериментов на НС переменной структуры была получена двухслойная сеть. Ошибка для данной структуры без учета артефактов составила 8% на обучающей выборке, и 5% на тестовой выборке. Ошибка для данной структуры с учетом артефактов составила 4% и на обучающей и на тестовой выборках.

Таким образом, эксперименты показали, что наиболее подходящей структурой для задачи экстраполяции числа звонков Call-центра является нейронная сеть переменной структуры.

#### Литература

1. Томашевич Д.С., Томашевич Н.С., Галушкин А.И., Муромский М.Ю. Учет артефактов при формировании обучающей выборки для задачи экстраполяции функции. Сб. Докл. VI Всероссийской конференции "Нейрокомпьютеры и их применение", Москва, 16-18 февраля, 2000.
2. Томашевич Д.С., Томашевич Н.С., Галушкин А.И., Муромский М.Ю. Применение нейронных сетей в Call-центрах. Сб. Докл. VI Всероссийской конференции "Нейрокомпьютеры и их применение", Москва, 16-18 февраля, 2000.
3. Томашевич Д.С., Томашевич Н.С., Галушкин А.И., Муромский М.Ю. Решение задачи экстраполяции неотрицательной функции на нейронной сети фиксированной структуры с полными последовательными связями. Сб. Докл. VI Всероссийской конференции "Нейрокомпьютеры и их применение", Москва, 16-18 февраля, 2000.
4. Галушкин А.И., Томашевич Н.С., Томашевич Д.С. Обоснование необходимости введения обратных связей и решение задачи экстраполяции неотрицательной функции на трехслойной нейронной сети фиксированной структуры с обратными связями. Сб. Докл. VI Всероссийской конференции "Нейрокомпьютеры и их применение", Москва, 16-18 февраля, 2000.
5. Томашевич Н.С., Томашевич Д.С. Решение задачи экстраполяции неотрицательной функции на нейронной сети переменной структуры. Сб. Докл. VI Всероссийской конференции "Нейрокомпьютеры и их применение", Москва, 16-18 февраля, 2000.
6. Томашевич Н.С., Томашевич Д.С. О выборе шага обучения нейронной сети с полными последовательными связями. Сб. Докл. VI Всероссийской конференции "Нейрокомпьютеры и их применение", Москва, 16-18 февраля, 2000.



ABOUT ONE PROBLEM OF CALL-CENTER MANAGEMENT USING NEURAL NETWORKS

Tomashevich N.S., Tomashevich D.S., Galushkin A.I., Muromsky M.U., Shachnev E.A.

Scientific Center of Neurocomputers  
107066, Moscow, st. Novaya Basmannaya, 20

Investigation of different neural network frames for decision of problem of calls number of one Call-center extrapolation taking into account and without taking into account artifacts is carried out. Defferent frames of neural networks are considered: fixed frame with full consecutive connections, fixed frame with feedbacks, variable frame. Experimental results have shown superiority of variable frame for decision of the given problem.

**Introduction**

Call-centers are organizations which are engaged in professional processing of telephone calls. Agents of Call-center immediately interact with clients. Each agent have any skill or several skills. Skill is understood as a knack of agent to take an order on pizza, airplane tickets and so on. Teaching of agent and getting by him of any skill is performed at the expenses of Call-center and costs sufficiently expensive.

One of the important tasks arising to Call-center is the correct allocation of manpower so that any agent didn't stand without work, and at the same time any client's call wasn't lost. This optimization problem includes following complicated subtask: it is required to form the forecast of calls number for one month forward so that manager of Call-center could correctly allocate a manpower.

Description of different methods for construction of neural networks of different frames for decision of the problem of calls number of one Call-center extrapolation is adduced. At decision of this problem authors were used database of one Call-center, containing datas about calls number during one year with quantization interval equal to half-hour, and also datas about artifacts, which are different events which can somehow influence on calls number.

**Neural network problem definition**

Calls number from database of one Call-center represents itself a time series specified on interval  $[1, M]$ . It is known that  $A$  different kinds of artifacts can influence on behavior of this time series, at that arbitrary number of artifacts in arbitrary combinations can happen in every point of time  $n \in \{1, \dots, M\}$ .

Since we decide an extrapolation problem, let's form learning sample by observing interval of time series of length  $T$  and by predicting  $\alpha$  following values of this time series. That is in each point of time  $n \in \{T, \dots, M - \alpha\}$  an example of learning sample is generated, in which the *input signal* of neural network consists of:

- vector of function  $F$  values (of length  $T$ ) from observation interval  $[n - T + 1, n]$ ;
- $(T + \alpha)$  vectors of artifacts (each of length  $A$ ) consisting of 0 and 1. If  $i$ -th kind of artifact,  $i \in \{1, \dots, A\}$ , presents on interval  $[n - T + 1, n + \alpha]$  in  $j$ -th point of time,  $j \in \{1, \dots, T + \alpha\}$ , then the  $i$ -th item of  $j$ -th vector is setting up equal to 1, otherwise it is equal to 0.

And *desired output signal* is the vector of function  $F$  values on interval  $[n + 1, n + \alpha]$  of length  $\alpha$ . Hence, the number of learning examples is equal to  $M - T - \alpha$ .

*Output signal* of neural network in point of time  $n \in \{T, \dots, M - \alpha\}$  is the vector (of length  $\alpha$ ) consisted of preceded by neural network function  $F$  values on interval  $[n + 1, n + \alpha]$ , denoted by  $\bar{y}(n) = \{y_1(n), \dots, y_\alpha(n)\}$ .

*Decision error* is defined as difference between desired and real output signal of neural network in discrete point of time  $n \in \{T, \dots, M - \alpha\}$ , that is decision error shows the length up to right value. In our case decision error in point of time  $n$  represents itself vector, which  $j$ -th item  $e_j(n)$  is equal to:

$$e_j(n) = y_j(n) - F(n + j), \tag{2}$$

where  $n$  – current point of time,  $n \in \{T, \dots, M - \alpha\}$ ;

$j$  – current position on extrapolation interval,  $j \in \{1, \dots, \alpha\}$ .

*Activation function* must be monotonic with real-valued output signal. In the problem decided in this paper function  $F(x)$  takes on only non-negative values. Size of function values changing is  $[0, 2m]$ , where  $2m$  is the maximum value of function  $F$  on interval  $[1, M]$ . So, let's use the following activation function of arctangent type:

$$y_{h_k}(n) = m \left( \frac{2}{\pi} \arctg(g_{h_k}(n)) \right) + m, \tag{3}$$

where  $y_{h_k}(n)$  - output of  $h_k$ -th neuron of  $k$ -th network layer,  $k \in \{1, \dots, K\}$ , in point of time  $n$ ;

$g_{h_k}(n)$  - activation of  $h_k$ -th neuron of  $k$ -th network layer,  $k \in \{1, \dots, K\}$ , in point of time  $n$ .

**Optimization functional**  $\Phi(n)$  was taken the same for all neural network frames, namely, minimizing mean-square error. At learning on  $n$ -th example,  $n \in \{T, \dots, M - \alpha\}$ , the functional is computed as:

$$\Phi(n) = \min \frac{1}{2} \sum_{h_K=1}^{\alpha} e_{h_K}^2(n), \quad (4)$$

where  $e_{h_K}(n)$  - error of  $h_K$ -th neuron of output layer, computed by formula (2).

Search of extremum of optimization functional for all neural network frames was carried out by **gradient method with random choice of initial conditions**. The gist consists in that the initial weights values are chosen in a random way, and then at network learning after  $n$ -th iteration,  $n \in \{T, \dots, M - \alpha\}$ , of each epoch the error is computed, and all network weights are adjusted in following way:

$$w_{h_k, h_{(k-1)}}(n+1) = w_{h_k, h_{(k-1)}}(n) - K_k^* \frac{d\Phi(n)}{dw_{h_k, h_{(k-1)}}}, \quad (5)$$

where  $K_k^*$  - is a coefficient of learning velocity of  $k$ -th network layer, which determine width of step on the error surface.

**Construction of multilayer neural network of fixed frame with full consecutive connections**

Frame of  $K$ -layer network with full consecutive connections is shown in fig. 1 and it is described by equation (3) and following equation:

$$g_{h_k}(n) = \sum_{h_{(k-1)=0}^{H_{(k-1)}} w_{h_k, h_{(k-1)}} \cdot y_{h_{(k-1)}}(n), \quad (6)$$

where  $H_{(k-1)}$  - number of neurons of  $(k-1)$ -th layer; when  $(k-1) = 0$ ,  $H_0 = T$ ;

$y_{h_{(k-1)}}(n)$  - output signal of  $h_{(k-1)}$ -th neuron of  $(k-1)$ -th NN layer; when  $h_{(k-1)} = 0$ ,  $y_0(n) = 1$ ; when  $(k-1) = 0$  and  $h_0 > 0$ ,  $y_{h_0}(n) = F(n - h_0 + 1)$ ;

$w_{h_k, h_{(k-1)}}$  - weight of connection from  $h_{(k-1)}$ -th neuron of  $(k-1)$ -th layer to  $h_k$ -th neuron of  $k$ -th layer.

Number of adjusted parameters of  $K$ -layer neural network is equal to:

$$\sum_{k=1}^K (H_{(k-1)} + 1) \cdot H_k, \quad (7)$$

where  $H_{(k-1)}$  - number of neurons of  $(k-1)$ -th layer; when  $(k-1) = 0$ ,  $H_0 = T$ ;

$H_k$  - number of neurons of  $k$ -th NN layer.

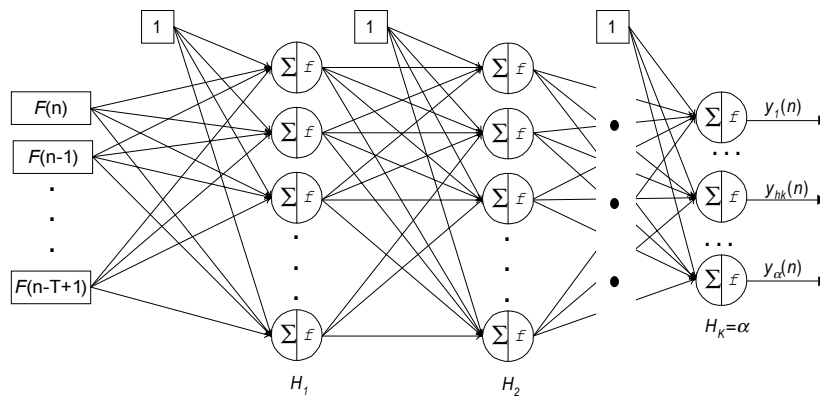


Fig. 1. Frame of  $K$ -layer network with full consecutive connections



**Construction of three-layer neural network of fixed frame with full consecutive connections and feedbacks**

Frame of three-layer neural network with full consecutive connections and feedbacks is shown in fig. 2 and it is described by equation (3) and following equations:

Activation of  $h_1$ -th neuron of 1-st network layer  $g_{h_1}(n)$  in point of time  $n \in \{T, \dots, M - \alpha\}$  is computed by formula:

$$g_{h_1}(n) = \left( w_{h_1,0} + \sum_{i=1}^T w_{h_1,i} \cdot F(n-i+1) \right) + \sum_{k=1}^3 \sum_{d_{k,1}=1}^{D_{k,1}} \sum_{h_k^{d_{k,1}}=1}^{H_k} w_{h_1,h_k^{d_{k,1}}} \cdot y_{h_k^{d_{k,1}}}(n), \quad (8)$$

where  $k$  – index by numbers of layers,  $k = \{1,2,3\}$ ;

$d_{k,1}$  - number of delay block, which delays output values of  $k$ -th layer before their feeding on input of 1-st layer;

$D_{k,1}$  - size of memory bank (or maximum number of delay blocks) at feeding of output values of  $k$ -th,  $k = \{1,2,3\}$ , layer on input of 1-st NN layer;

$H_k$  - number of neurons of  $k$ -th network layer,  $k = \{1,2,3\}$ ;

$h_k^{d_{k,1}}$  -  $h_k$ -th item of  $d_{k,1}$ -th delay block,  $k = \{1,2,3\}$ ;

$w_{h_1,h_k^{d_{k,1}}}$  - weight from  $h_k$ -th item of  $d_{k,1}$ -th delay block,  $k = \{1,2,3\}$ , to  $h_1$ -th neuron of 1-st network layer,  $h_1 \in \{1, \dots, H_1\}$ ;

$w_{h_1,i}$  - weight from  $i$ -th input to  $h_1$ -th neuron of 1-st NN layer;

$y_{h_k^{d_{k,1}}}(n)$  - output of  $h_k$ -th item of  $d_{k,1}$ -th delay unit,  $k = \{1,2,3\}$ , in point of time  $n$ .

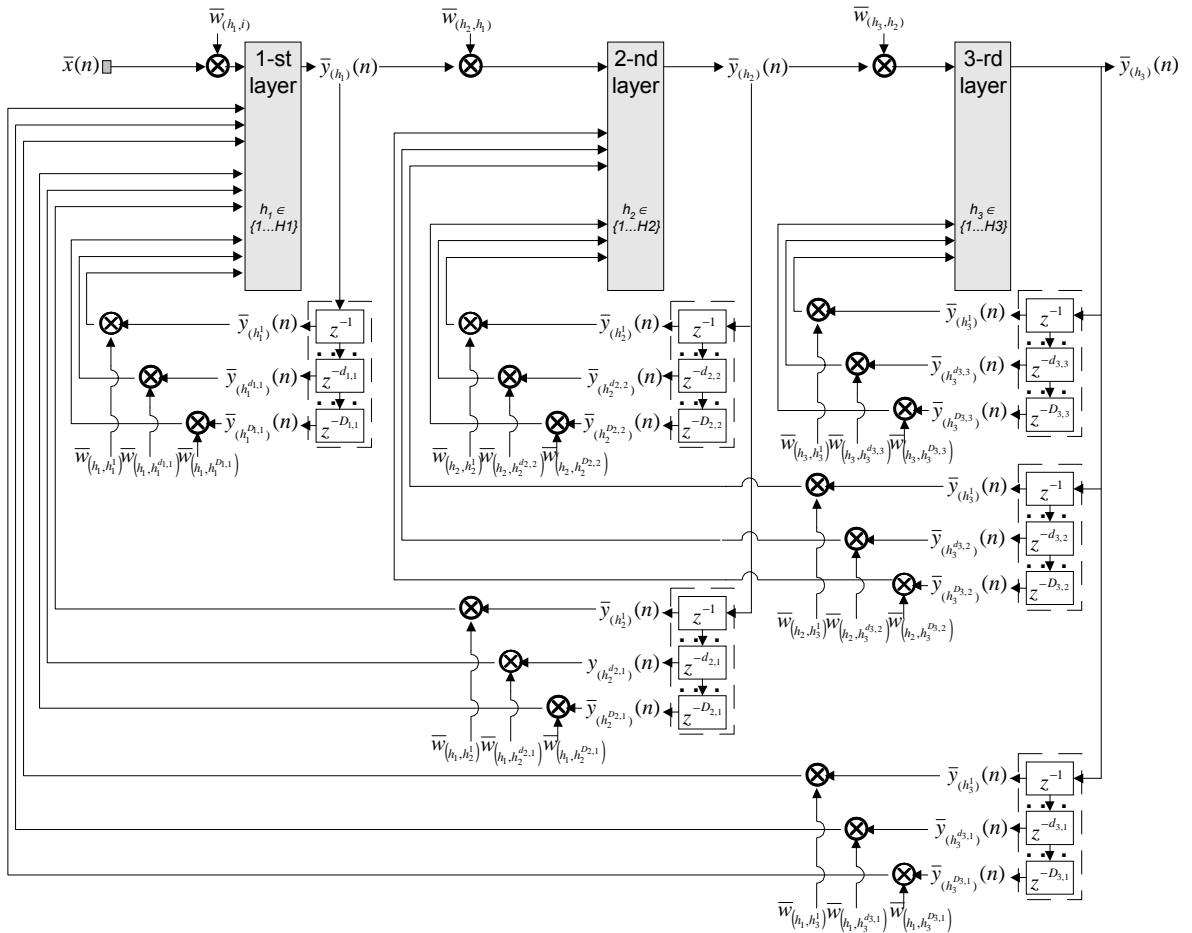


Fig. 2. Frame of three-layer neural network with full consecutive connections and feedbacks

Activation of  $h_2$ -th neuron of 2-nd network layer  $g_{h_2}(n)$  in point of time  $n \in \{T, \dots, M - \alpha\}$  is computed by formula:

$$g_{h_2}(n) = \sum_{h_1=0}^{H_1} w_{h_2, h_1} \cdot y_{h_1}(n) + \sum_{k=2}^3 \sum_{d_{k,2}=1}^{D_{k,2}} \sum_{h_k^{d_{k,2}}=1}^{H_k} w_{h_2, h_k^{d_{k,2}}} \cdot y_{h_k^{d_{k,2}}}(n), \quad (9)$$

where  $d_{k,2}$  - number of delay block, which delays output values of  $k$ -th layer before their feeding on input of 2-nd layer;

$D_{k,2}$  - size of memory bank (or maximum number of delay blocks) at feeding of output values of  $k$ -th,  $k = \{2,3\}$ , layer on input of 2-nd NN layer;

$h_k^{d_{k,2}}$  -  $h_k$ -th item of  $d_{k,2}$ -th delay block,  $k = \{2,3\}$ ;

$w_{h_2, h_k^{d_{k,2}}}$  - weight from  $h_k$ -th item of  $d_{k,2}$ -th delay block,  $k = \{2,3\}$ , to  $h_2$ -th neuron of 2-nd network layer,  $h_2 \in \{1, \dots, H_2\}$ ;

$w_{h_2, h_1}$  - weight from  $h_1$ -th neuron of 1-st NN layer to  $h_2$ -th neuron of 2-nd NN layer;

$y_{h_k^{d_{k,2}}}(n)$  - output of  $h_k$ -th item of  $d_{k,2}$ -th delay block,  $k = \{2,3\}$ , in point of time  $n$ .

Activation of  $h_3$ -th neuron of 3-rd network layer  $g_{h_3}(n)$  in point of time  $n \in \{T, \dots, M - \alpha\}$  is computed by formula:

$$g_{h_3}(n) = \sum_{h_2=0}^{H_2} w_{h_3, h_2} \cdot y_{h_2}(n) + \sum_{d_{3,3}=1}^{D_{3,3}} \sum_{h_3^{d_{3,3}}=1}^{H_3} w_{h_3, h_3^{d_{3,3}}} \cdot y_{h_3^{d_{3,3}}}(n), \quad (10)$$

where  $d_{3,3}$  - number of delay block, which delays output values of 3-rd layer before their feeding on input of 3-rd layer;

$D_{3,3}$  - size of memory bank (or maximum number of delay blocks) at feeding of output values of 3-rd layer on input of 3-rd NN layer;

$h_3^{d_{3,3}}$  -  $h_3$ -th item of  $d_{3,3}$ -th delay block;

$w_{h_3, h_3^{d_{3,3}}}$  - weight from  $h_3$ -th item of  $d_{3,3}$ -th delay block to  $h_3$ -th neuron of 3-rd network layer,  $h_3 \in \{1, \dots, H_3\}$ ;

$w_{h_3, h_2}$  - weight from  $h_2$ -th neuron of 2-nd NN layer to  $h_3$ -th neuron of 3-rd NN layer;

$y_{h_3^{d_{3,3}}}(n)$  - output of  $h_3$ -th item of  $d_{3,3}$ -th delay block in point of time  $n$ .

#### **Construction of multilayer neural network of variable frame**

Frame of network is named «variable», because the number of neurons and number of layers of network are unknown in advance, and they are determined in process of network creation. After creation the network has the same frame as network in fig. 1. It is impossible to say beforehand how much layers and neurons will be in NN. This is determined in process of network learning and depends on complexity of approximated function  $F$ . Gist of network with variable frame creation is following.

Firstly, we build network with only one neuron in first layer and  $\alpha$  neurons in second (output) layer. This frame represents itself the learning block, shown in fig. 3. For this learning block some number of random choices  $N_{choice}$  of initial weights values, before defined by user, is performed. After each random choice the NN error is computed. When all choices are done, weights values for learning block, which correspond to minimum error, are selected. Then learning block teaches  $N_{epoch}$  epochs.

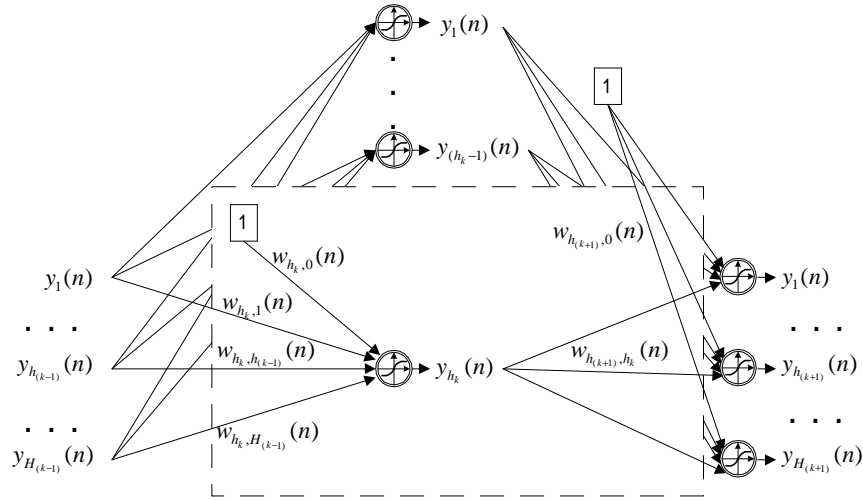


Fig. 3. Learning block of  $h_k$ -th neuron

Let's name the described above process by **block's learning phase** (this phase consists of  $N_{choice}$  random choices of initial weights values and of  $N_{epoch}$  learning epochs).

Then one more neuron and connections from this neuron to neurons of second layer are added in first layer. Now learning block consists of second neuron of first layer (with all input and output connections) and of  $\alpha$  neurons of second layer, but only with connections from second neuron of first layer and with bias. Block's learning phase is carried out for this neuron, too.

Similarly, neurons are added to the first layer. At that each neuron must improve approximation quality of time series by neural network. Approximation quality is appreciated using error, averaged over total learning sample and over all outputs:

$$E = \frac{1}{M - \alpha - T} \frac{1}{\alpha} \sum_{n=T}^{M-\alpha} \sum_{h_k=1}^{\alpha} |e_{h_k}(n)|, \quad (11)$$

where  $e_{h_k}(n)$  - error of  $h_k$ -th neuron of output  $K$ -th NN layer on  $n$ -th example.

If the next added to first layer neuron doesn't improve approximation quality, then phase of block's learning is repeated. Let's denote the maximum number of block's learning phase repeats for one neuron by  $N_{repeat}$ . If after  $N_{repeat}$  repeats the current neuron doesn't improve approximation quality (that is error of current neuron is not less than error of previous neuron), then current neuron is removed, and new third layer of  $\alpha$  neurons is added to neural network. Now we choose in a random way only weights between  $\alpha$  neurons of 2-nd layer and  $\alpha$  neurons of 3-rd layer. At the same time  $N_{choice}$  random choices of initial weights values is accomplished, then  $N_{epoch}$  learning epochs is carried out, and if added layer doesn't improve approximation quality, then described above process is repeated at most  $N_{repeat}$  times. If after  $N_{repeat}$  repeats 3-rd layer doesn't improve approximation quality, then it is removed and we draw conclusion that with given values of  $N_{choice}$ ,  $N_{epoch}$  and  $N_{repeat}$  it is impossible to achieve the better approximation quality. If 3-rd layer improves approximation quality, then we continue to add neurons to the 2-nd layer in the same way, as to the 1-st layer.

Similarly, we create so much layers as it is necessary to achieve the required approximation quality of time series.

To decrease NN frame and to improve approximating capabilities of each neuron, it is useful to introduce parameter  $\Delta_{best}$ , which shows on how much each neuron must improve total approximation quality of function  $F$  by neural network.

### Experimental results and conclusions

Experiments on described above three frames of neural networks were carried out. At that for each NN frame two kinds of experiments were carried out: taking into account and without taking into account artifacts of learning sample. Following results were achieved.

Learning error on tree-layer NN of fixed frame with full consecutive connections without taking into account artifacts was equal to 10% from maximum function value, and test error – 5%. Learning error on the same frame, but with taking into account artifacts, was equal to 6%, and test error – 12%. Such increasing of test error is caused by

adding to network of very big quantity of weights from artifact inputs (total number of different kinds of artifacts is 20, value of observation interval  $T$  is 30 days, that is number of artifact weights is equal to 600). Since number of one kind of artifacts in time series wasn't exceeding 5, it is understood, that weights from artifact inputs couldn't be taught on such small number of examples, and resulted noisiness of output signal of neural network.

Experiments on three-layer NN of fixed frame with full consecutive connections and feedbacks was carried out without taking into account artifacts, both learning and test error was equal approximately to 25%. Such sharp increasing of errors was caused by introduction of big quantity of weights, which are necessary to realization of feedback with at least one delay unit. Since authors didn't know how to right choice initial values of feedback weights, and choice of weights values in a random way results a big error, then experiments on this frame with taking into account artifacts wasn't carried out.

As a result of experiments on NN of variable frame we received two-layer network. Error for this frame without taking into account artifacts on learning sample was 8% and on test sample 5%. Error for this frame with taking into account artifacts was 4% for both learning and test samples.

Thereby, experimental results have shown, that neural network of variable frame is the most suitable frame for decision of problem of calls number of Call-center extrapolation.

#### *References*

1. Tomashevich D.S., Tomashevich N.S., Galushkin A.I., Muromsky M.U. Artifacts taking into account at generation of learning sample for problem of function extrapolation. Proc. of VI All-Russian Conf. "Neurocomputers and their applications", Moscow, 16-18 February, 2000.
2. Tomashevich D.S., Tomashevich N.S., Galushkin A.I., Muromsky M.U. Application of neural networks in Call-centers. Proc. of VI All-Russian Conf. "Neurocomputers and their applications", Moscow, 16-18 February, 2000.
3. Tomashevich D.S., Tomashevich N.S., Galushkin A.I., Muromsky M.U. Decision of problem of non-negative function extrapolation using fully feedforward-connected neural network of fixed frame. Proc. of VI All-Russian Conf. "Neurocomputers and their applications", Moscow, 16-18 February, 2000.
4. Galushkin A.I., Tomashevich N.S., Tomashevich D.S. Justification of necessity of feedbacks introduction and decision of problem of non-negative function extrapolation using tree-layer neural network of fixed frame with feedbacks. Proc. of VI All-Russian Conf. "Neurocomputers and their applications", Moscow, 16-18 February, 2000.
5. Tomashevich N.S., Tomashevich D.S. Decision of problem of non-negative function extrapolation using neural network of variable frame. Proc. of VI All-Russian Conf. "Neurocomputers and their applications", Moscow, 16-18 February, 2000.
6. Tomashevich N.S., Tomashevich D.S. About adjusting of learning rate for fully feedforward-connected neural network of fixed frame. Proc. of VI All-Russian Conf. "Neurocomputers and their applications", Moscow, 16-18 February, 2000.