

## МЕТОД СИНТЕЗА ПАРАМЕТРОВ МАТЕМАТИЧЕСКОЙ МОДЕЛИ СВЕРТОЧНОЙ НЕЙРОННОЙ СЕТИ С РАСШИРЕННЫМ ОБУЧАЮЩИМ МНОЖЕСТВОМ

Немков Р.М.

*ФГАОУ ВПО «Северо-Кавказский федеральный университет», Ставрополь, Россия (355009, г. Ставрополь, ул. Пушкина, 1), e-mail: nemkov.roman@yandex.ru*

В настоящей статье предлагается алгоритм изменения рецептивных полей для сверточных нейронных сетей на произвольной комбинации сверточных слоёв. На основе этого алгоритма разработан метод синтеза параметров математической модели сверточной сети с расширенным обучающим множеством, генерируемым изменением её рецептивных полей. Построена математическая модель сверточной нейронной сети (СНС) с расширенным обучающим множеством и разработан алгоритм синтеза её параметров, который обеспечивает адаптацию алгоритма обратного распространения ошибки к изменению формы рецептивных полей перед подачей очередного паттерна в процессе обучения сети. Проведены эксперименты по оценке обобщающей способности СНС на выборках, используемых для распознавания объектов, проведена оценка качества распознавания на реальных объектах с использованием стенда «Мехатроника» НПО «Андроидная техника».

Ключевые слова: сверточные нейронные сети, распознавание образов, рецептивные поля, обучающее множество.

## SYNTHESIS METHOD OF MATHEMATICAL MODEL PARAMETERS OF THE CONVOLUTIONAL NEURAL NETWORK WITH EXTENDED TRAINING SET

Nemkov R.M.

*Federal State Autonomous Educational Institution for Higher Professional Education «North-Caucasus Federal University», Stavropol, Russia (355009 Russia, Stavropol, Pushkin street, 1), e-mail: nemkov.roman@yandex.ru*

In this paper we propose algorithm of receptive fields changes for convolutional neural networks (CNN) on any convolutional layers combination. On the basis of this algorithm we developed the synthesis method of parameters of convolutional network mathematical model with extended training set generated by the change of its receptive fields. The mathematical model of convolutional neural network with extended training set was made. The algorithm of parameters synthesis of this model was developed. This algorithm ensures an adaptation of back-propagation algorithm to the change of receptive fields form before the analysis of another pattern in the process of the network training. We conducted the experiments on the estimation of generalization ability of the proposed model. The training sets such as MNIST, Small NORB were used for these experiments. Also experiments were conducted for an estimation of quality of network recognition process with the use of real objects. We made these experiments on the stand with video camera called «Mechatronics» (SPA «Android technics»).

Keywords: convolutional neural networks, pattern recognition, receptive fields, training set.

Инвариантное распознавание изображений является ключевым направлением в области распознавания образов. В настоящее время для инвариантного распознавания изображений наилучшие результаты показывают стохастические модели, а именно подкласс искусственных нейронных сетей (ИНС) – сверточные нейронные сети (СНС) [8]. Однако обучение ИНС – это плохо обусловленная (некорректная) обратная задача [5]. Плохая обусловленность означает, что даже большой набор данных может нести в себе малый объем информации о решении задачи. Поэтому ключевая роль в процессе синтеза параметров конкретной математической модели ИНС принадлежит обучающим данным. Составление же репрезентативной обучающей выборки является одной из самых сложных задач в машинном обучении [1; 5].

Известно несколько альтернатив, применяемых вместо создания большой репрезентативной выборки, по расширению обучающего множества посредством добавления преобразованных паттернов.

1. Использование специальных алгоритмов, которые перед обучением расширяют обучающее множество [7]. Однако настройка, подбор алгоритмов и получаемого размера множества становятся очень трудным этапом.
2. Использование регуляризации [5]. Основной недостаток – более сложное обучение и трудность выработки устойчивости сети к требуемым искажениям.
3. Изменение внутренних параметров математической модели сети [2]: искажения, накладываемые на вектор антиградиента, на настраиваемые веса сети, временное исключение некоторых нейронов или связей между ними во время обучения сети. Это позволяет решать в основном прикладные задачи: препятствовать переобучению, вырабатывать устойчивость работы сети к возможным отказам некоторых элементов. Увеличение обобщающей способности – лишь косвенное следствие применения подобных решений.

Предлагается за счёт изменения таких внутренних параметров СНС, как форма рецептивных полей (РП) [2], создавать искажения по отношению к текущему паттерну, тем самым получая новые паттерны и расширяя обучающую выборку [4; 9]. Для реализации предложенного подхода необходимо разработать алгоритм изменения формы РП для различных комбинаций сверточных слоёв СНС, а также метод синтеза параметров математической модели СНС с расширенным обучающим множеством, генерируемым изменением её РП.

Известный обобщённый алгоритм синтеза параметров для СНС включает в себя следующие шаги:

1. Подача и нормировка паттерна на вход СНС.

$$(\text{Вход}_{m,n}^i)^{(\text{new})} = \frac{2 * (\text{Вход}_{m,n}^i)^{(\text{old})}}{255} - 1, \quad (1)$$

где  $(\text{Вход}_{m,n}^i)^{(\text{new})}$  - новое значение входа нейрона, расположенного на карте  $i$  в позиции  $m, n$  после линейной нормировки.

2. Осуществление прямого прогона, т.е. распространение вычисляемых сигналов от входного слоя к выходному: на сверточном слое (С-слой) (2), на слое усреднения (S-слой) (3), выходном слое (4, 5) соответственно.

$$y_{m,n} = C_{m,n}^i = \varphi(p) = \varphi(b + \sum_{q \in Q_i} \sum_{k=0}^{K_C-1} \sum_{l=0}^{K_C-1} X_{m+k,n+l}^q * \text{Ядро}_{k,l}^q), \quad (2)$$

$$y_{m,n} = S_{m,n}^i = \varphi(p) = \varphi(b + u \sum_{k=0}^{K_S-1} \sum_{l=0}^{K_S-1} C_{m^*K_S+k,n^*K_S+l}^i), \quad (3)$$

где  $C_{m,n}^i$ ,  $S_{m,n}^i$  – выход нейрона, расположенного на  $i$ -й карте  $S$  или  $S$ -слоя в позиции  $m$ ,  $n$ ,  $\varphi(\cdot) = A \cdot \tanh(B \cdot p)$  при  $A=1.7159$ ,  $B=2/3$ ,  $p$  – взвешенная сумма,  $b$  – смещение,  $u$  – настраиваемый параметр для каждой  $S$ -карты (усредняющий коэффициент),  $Q_i$  – множество индексов карт предыдущего слоя, связанных с картой  $C^i$ ,  $K_C$ ,  $K_S$  – размер квадратного РП для нейрона  $C_{m,n}^i$  и  $S_{m,n}^i$  соответственно,  $X_{m+k,n+l}^q$  – входное значение для нейрона  $C_{m,n}^i$ ,  $Ядро_{k,l}^q$  –  $q$ -я часть настраиваемых параметров, которая отвечает за взаимодействие с  $q$ -й картой предыдущего слоя.

$$p_i = \left( \sum_{k=1}^K C_k * w_{i,k} \right) + b_i, \quad (4)$$

$$Output_i = \varphi(p_i) = \frac{\exp(p_i)}{\sum_{k=1}^{SizeOutput} \exp(p_k)}, \quad (5)$$

где  $i = 1..SizeOutput$ ,  $SizeOutput$  – кол-во нейронов в выходном слое,  $K$  – кол-во нейронов в слое, предшествующем выходному слою (вырожденный  $S$ -слой),  $w_{i,k}$  – настраиваемые параметры выходного слоя,  $b_i$  – смещение для нейрона этого слоя,  $p_i$  – взвешенная сумма нейронов этого слоя,  $Output_i$  – выход  $i$ -го нейрона выходного слоя.

3. Вычисление ошибки на выходном слое.

$$E = - \sum_{k=1}^{SizeOutput} t_k * \log(Output_k), \quad (6)$$

где  $t_k \in [0,1]$  –  $k$ -й элемент вектора-учителя.

4. Осуществление обратного прогона.

Для выходного слоя (7), для вырожденного слоя (8), для  $S$ -слоёв (9, 10), для  $S$ -слоёв (11, 12), для смещений (13) соответственно.

$$\frac{\partial E}{\partial w_{i,j}^\lambda} = \delta_i^\lambda * y_j^{\lambda-1}, \quad (7)$$

где  $\delta_i^\lambda = \frac{\partial E}{\partial p_i^\lambda} = Output_i - t_i$  – локальный градиент (невязка), получаемая на выходном

слое, который далее распространяется от выхода к входу,  $\lambda$  – номер слоя,  $w_{i,j}^\lambda$  – настраиваемые параметры для  $\lambda$  слоя,  $y_j^{\lambda-1}$  – выходное значение нейрона  $j$  со слоя  $\lambda-1$ .

$$\frac{\partial E}{\partial w_{i,j}^{\lambda-1}} = \sum_{k \in D} \delta_k^\lambda * w_{k,i}^\lambda * \varphi'(p_i^{\lambda-1}) * y_j^{\lambda-2} = y_j^{\lambda-2} * \left( \sum_{k \in D} \delta_k^\lambda * w_{k,i}^\lambda \right) * \varphi'(p_i^{\lambda-1}) = \delta_i^{\lambda-1} * y_j^{\lambda-2}, \quad (8)$$

где  $D$  – множество нейронов из последующего слоя ( $\lambda$ -слой), соединённых с нейроном  $y_i^{\lambda-1}$ ,  $\varphi'(\cdot)$  производная от  $\varphi(\cdot)$ ,

Для С-слоя и S-слоя вычисление компонентов вектора градиента происходит в два этапа: сначала вычисляется накопленная невязка, затем сам компонент вектора градиента по параметру сети.

$$\delta_{m,n}^{\lambda} = \sum_{i \in \bigcup_{z=1}^N D_z} \delta_i^{\lambda+1} * w_i^{\lambda+1}[m,n] * \varphi'(p_{m,n}^{\lambda}), \quad (9)$$

где  $\delta_{m,n}^{\lambda}$  – невязка, собираемая для нейрона с координатами  $m, n$  в пределах карты слоя  $\lambda$ ,  $w_i^{\lambda+1}[m,n]$  – настраиваемый параметр для связи, соединяющий нейрон  $m, n$  на S-слое и нейрон с невязкой  $\delta_i^{\lambda+1}$ ,  $w_i^{\lambda+1}[m,n] = \text{Ядро}_i^{\lambda+1}[\Delta t, \Delta z]$ ,  $(\Delta t, \Delta z)$  – смещение в рамках ядра для настраиваемого параметра связи,  $N$  – кол-во карт, которые связаны с текущим нейроном из последующего слоя,  $D_z$  – множество связанных нейронов последующего слоя из  $z$ -й карты, которые связаны с нейроном  $m, n$ .

$$\frac{\partial E}{\partial u^{\lambda}} = \sum_{k=0}^{K_S} \sum_{l=0}^{K_S} \frac{\partial E}{\partial u_{k,l}^{\lambda}}, \quad (10)$$

где  $\frac{\partial E}{\partial u_{k,l}^{\lambda}} = \sum_{m=0}^{SizeS} \sum_{n=0}^{SizeS} \delta_{m,n}^{\lambda} * y_{m*K_S+k, n*K_S+l}^{\lambda-1}$ , т.к.  $u_{k,l}^{\lambda} = u^{\lambda}$ ,  $SizeS$  – размер карты S-слоя.

$$\delta_{m,n}^{\lambda} = \sum_{i \in D} \delta_i^{\lambda+1} * w_i^{\lambda+1}[m,n] * \varphi'(p_{m,n}^{\lambda}), \quad (11)$$

где  $D$  – множество нейронов на последующей карте ( $\lambda+1$  слой), связанных с нейроном  $m, n$ ,  $w_i^{\lambda+1}[m,n]$  – это  $u_i^{\lambda+1}$  той карты S-слоя, с которой связана карта С-слоя.

$$\frac{\partial E}{\partial (\text{Ядро}_{k,l}^{\lambda})^q} = \sum_{m=0}^{SizeC} \sum_{n=0}^{SizeC} \delta_{m,n}^{\lambda} * y_{m+k, n+l}^{\lambda-1}, \quad (12)$$

где  $q$  – та часть ядра настраиваемых параметров, для которых получают компонент градиента,  $SizeC$  – размер карты С-слоя.

Обновления смещений для S-слоёв и С-слоёв происходят по формуле (13)

$$\frac{\partial E}{\partial b_i^{\lambda}} = \sum_{m=0}^{SizeCard} \sum_{n=0}^{SizeCard} \delta_{m,n}^{\lambda}, \quad (13)$$

где  $\frac{\partial E}{\partial b_i^{\lambda}}$  – компонент вектора градиента для смещения  $i$ -й S или С-карты слоя  $\lambda$ ,

$SizeCard$  – размер карты С-слоя или S-слоя, если происходит обновление смещения для С-слоя или S-слоя соответственно.

## 5. Корректировка настраиваемых параметров СНС.

$$w_{new} = w_{old} - \eta * \frac{\partial E}{\partial w_{old}}, \quad (14)$$

где  $w_{new}$ ,  $w_{old}$  – значения новых и старых параметров соответственно,  $\eta$  – скорость обучения,  $\frac{\partial E}{\partial w_{old}}$  – вектор градиента.

Известно, что классическая форма РП в СНС – это квадрат [8]. Для получения нестандартного РП предлагается использовать шаблон, элементами которого являются индексы [9], обозначающие своих соседей в пределах двух дискретных шагов от них на пиксельной матрице. При изменении всех РП, лежащих на карте, на настраиваемые параметры будет воздействовать дополнительная информация, что приведёт к выделению лучшего инварианта (рис. 1).

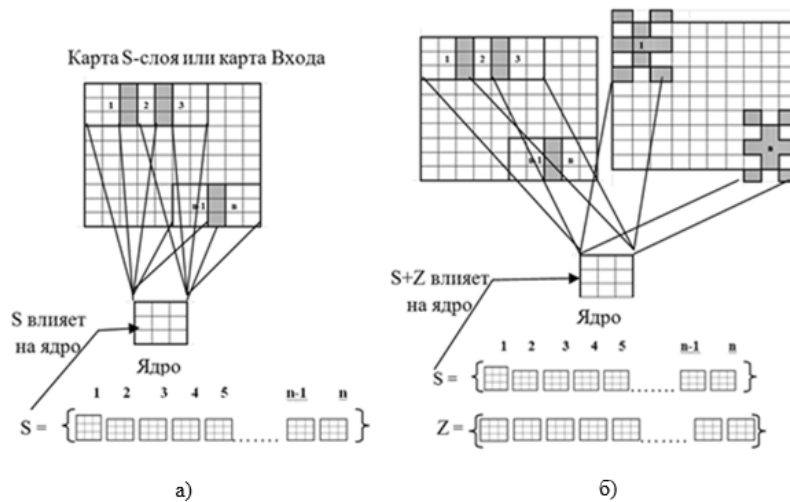


Рис. 1: а) стандартные РП, б) стандартные и добавочные РП

Вектор антиградиента, получаемый при использовании нестандартных РП, в контексте процесса обучения можно рассматривать как наложение искажений на вектор антиградиента, получаемый при использовании стандартных РП.

Использование РП нестандартной формы требует адаптации алгоритма обратного распространения ошибки на шаге 2 и 4: предлагается вместо формул (2) и (12) использовать формулы (15) и (17) соответственно.

$$y_{m,n} = C_{m,n}^i = \varphi(p) = \varphi(b + \sum_{q \in Q_i} \sum_{k=0}^{K_C-1} \sum_{l=0}^{K_C-1} X_{m+k+F_i(RP_{m,n};k;l), n+l+F_j(RP_{m,n};k;l)}^q * \text{Ядро}_{k,l}^q), \quad (15)$$

где  $F_i(RP_{m,n};k;l)$ ,  $F_j(RP_{m,n};k;l)$  – функции, возвращающие смещения по строке и столбцу для шаблона РП, принадлежащего нейрону  $m,n$  в позиции  $k, l$  внутри этого шаблона.  $index_{k,l}$  – есть элемент шаблона  $RP_{m,n}$  в позиции  $k, l$ ,  $index_{k,l} = 0..24$ . Данные функции определяются по следующим формулам:

$$F_i(\cdot) = \begin{cases} 0; & index_{k,l} \in \{0, 4, 5, 16, 17\} \\ 1; & index_{k,l} \in \{6, 7, 8, 18, 19\} \\ 2; & index_{k,l} \in \{20, 21, 22, 23, 24\} \\ -1; & index_{k,l} \in \{1, 2, 3, 14, 15\} \\ -2; & index_{k,l} \in \{9, 10, 11, 12, 13\} \end{cases} \quad F_j(\cdot) = \begin{cases} 0; & index_{k,l} \in \{0, 2, 7, 11, 22\} \\ 1; & index_{k,l} \in \{3, 5, 8, 12, 23\} \\ 2; & index_{k,l} \in \{13, 15, 17, 19, 24\} \\ -1; & index_{k,l} \in \{1, 4, 6, 10, 21\} \\ -2; & index_{k,l} \in \{9, 14, 16, 18, 20\} \end{cases}, \quad (16)$$

$$\frac{\partial E}{\partial (\text{Ядро}_{k,l}^\lambda)^q} = \sum_{m=0}^{SizeC} \sum_{n=0}^{SizeC} \delta_{m,n}^\lambda * y_{m+k+F_i(RP_{m,n}^\lambda; k;l), n+l+F_j(RP_{m,n}^\lambda; k;l)}^{\lambda-1}, \quad (17)$$

Предлагается алгоритм, изменяющий перед подачей паттерна РП у нейронов, лежащих на любой комбинации С-слоёв. Информация о таком изменении берётся из столбцов матрицы  $A_{9 \times L}$ .

$$L = N_0 + M, \quad (18)$$

где  $L$  – количество столбцов матрицы  $A$ ,  $N_0$  – количество паттернов в искомой обучающей выборке,  $M$  – количество добавленной информации о смене РП для преобразования  $M$  паттернов.

$$M(Q, T, V_j) = \sum_{j=1}^{V=QT} \left[ \frac{V_j}{100} * N_0 \right], \quad (19)$$

где  $Q$  – количество комбинаций С-слоёв, которые могут иметь или не иметь РП нестандартной формы. Использовались СНС с тремя С-слоями, поэтому конкретная комбинация представляет собой набор трёх цифр:  $Q_i = (x_1, x_2, x_3)$ ,  $x_j \in [0, 1]$ ,  $i=1..2^{(\text{Кол-во С-слоёв})}$ ,  $T$  – количество тактов,  $V_j$  – целое число, обозначающее процент от  $N_0$ ,  $0 \leq V_j \leq 100$ ,  $V$  – кол-во чисел  $V_j$ ,  $V=QT$ .

Каждый С-слой может иметь несколько типов РП:  $SetRP_i \in \{RP_1, RP_2, \dots, RP_{n_i}\}$ , где  $SetRP_i$  – множество типов РП для  $i$ -го С-слоя,  $RP_1$  – квадратное РП,  $n_i$  – количество типов РП для  $i$ -го С-слоя.

Для присвоения нейронам  $i$ -го С-слоя РП предлагается использовать два алгоритма. Алгоритм  $Alg_1(RP_i \in SetRP_i; C_i)$  присваивает всем нейронам слоя  $C_i$  РП с индексом  $RP_t$ ,  $t=1..T$  – номер такта,  $T=n_i$ . Такой алгоритм при соответствующих РП обеспечивает инвариантность к сдвигам, алгоритм  $Alg_2(RP \in SetRP_i; C_i)$  – присваивает каждому нейрону из слоя  $C_i$  случайное поле  $RP$  из  $SetRP_i$ . Такой алгоритм обеспечивает инвариантность к локально-аффинным преобразованиям типа *elastic distortions* [3]. Инвариантность к масштабу, текстурам фона, положению объекта, уровням освещённости обеспечивается исходной обучающей выборкой.

Поскольку свои РП могут иметь любые С-слои, то алгоритмы разметки слоя могут быть использованы алгоритмами разметки всех С-слоёв. Предлагается два таких алгоритма:

$$Strategy_1(Alg_1(\cdot), Q_j) = \begin{cases} Alg_1(RP_i; C_i), & \text{если } x_i=0 \\ Alg_1(RP_i; C_i), & \text{если } x_i=1 \end{cases} \quad (20)$$

$$Strategy_2(Alg_1(\cdot), Alg_2(\cdot), Q_j) = \begin{cases} Alg_1(RP_i; C_i), & \text{если } x_i=0 \\ Alg_2(RP \in SetRP_i; C_i), & \text{если } x_i=1 \end{cases} \quad (21)$$

где  $i=1\dots$  количество С-слоёв,  $x_i \in Q_j$ ,  $1 \leq t \leq T = n_i$ ,  $Q_j$  – конкретная комбинация С-слоёв, которые имеют или не имеют РП нестандартной формы.

Обобщённый алгоритм изменения формы РП для слоёв СНС представлен на рисунке

2.

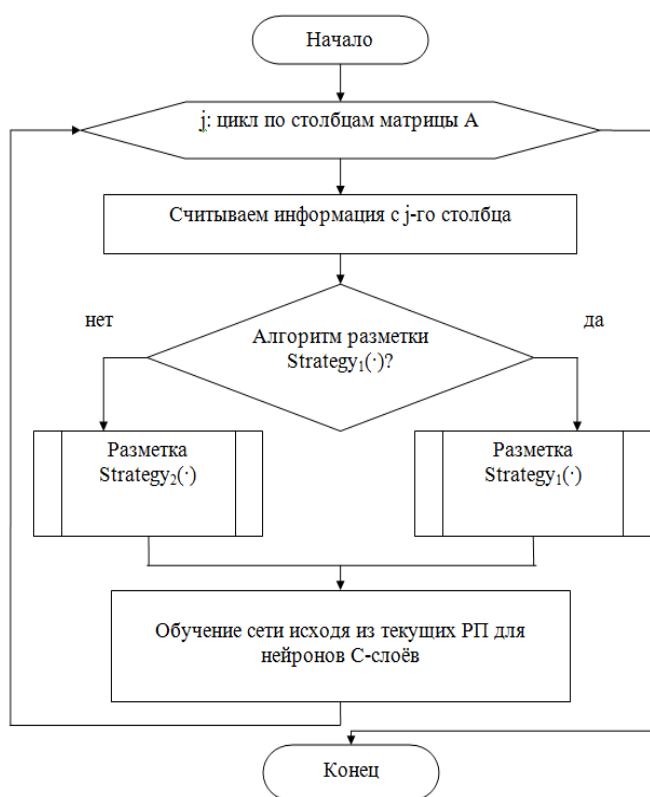


Рис. 2. Обобщённый алгоритм изменения формы рецептивных полей для слоёв СНС

Разработанный алгоритм синтеза параметров математической модели СНС с расширенным обучающим множеством представлен на рисунке 3.

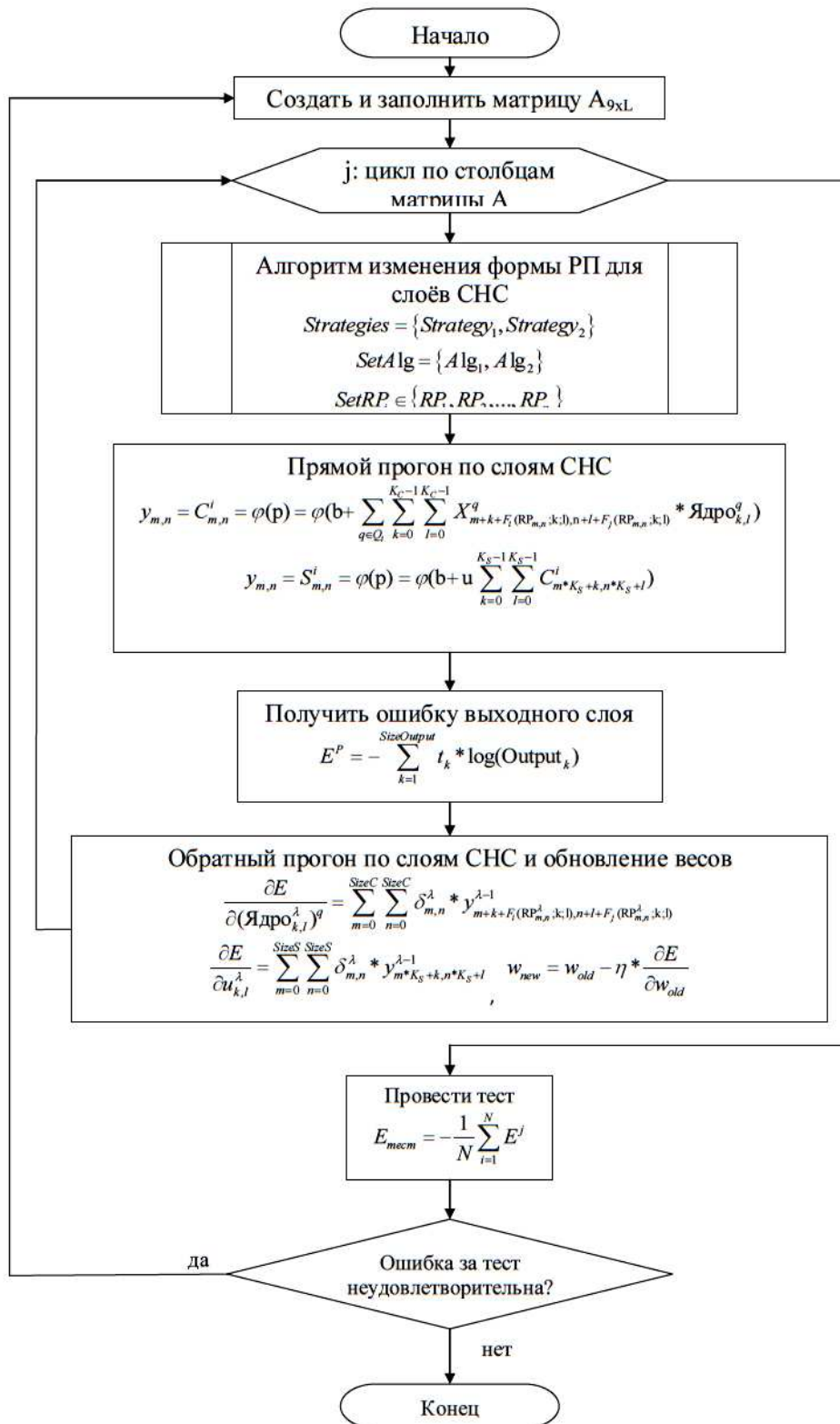


Рис. 3. Обобщённый алгоритм синтеза параметров математической модели СНС с расширенным обучающим множеством, генерируемым изменением её РП

Эксперименты по оценке обобщающей способности предлагаемой математической модели СНС с расширенным обучаемым множеством проводились с использованием трёх выборок: MNIST (рукописные цифры) [10], Small NORB (пять классов объектов) [11], выборка, созданная для распознавания 10 классов объектов со стенда «Мехатроника» [3].



Результаты вычислительных экспериментов представлены в таблице 1.

**Таблица 1**

Результаты вычислительных экспериментов по оценке обобщающей способности алгоритма обучения СНС предлагаемым способом и влияния редукции на процесс обучения

Достигнутая ошибка обобщения с использованием предлагаемой математической модели, %	Достигнутая ошибка обобщения без использования предлагаемой математической модели, %	Ошибка обобщения лучшего аналога, %	Время обучения без редукции, ч	Время обучения с редукцией, ч
MNIST				
0.6	2.8	0.23	46.15	31.2
Small NORB				
4.3	8.4	6.6	15.6	9.8
Выборка для стенда «Мехатроника»				
0.3	1.8	---	17.3	10.1

Из таблицы следует, что обучение СНС с помощью предлагаемого метода улучшает в среднем обобщающую способность сети на 2.5%. Численный метод редукции [4] расширенного обучающего множества позволяет сократить время обучения в среднем на 9 ч, что составляет 37% от первоначального времени обучения без редукции.

Для распознавания объектов со стенда «Мехатроника» [6] разработан программный комплекс, позволяющий распознавать объекты с камер мобильного робота. Комплекс состоит из двух модулей: выделение потенциально интересных мест на кадре и распознавание выделенных изображений, переданных на вход СНС. Оболочка комплекса и алгоритм выделения мест написаны на С#, распознавание с помощью СНС реализовано на С++ и подключается в виде библиотеки к основной программе.

Эксперименты с распознаванием объектов на стенде «Мехатроника» для различных расстояний от камеры (AXIS M1054) приведены в таблице 2.

**Таблица 2**

Процент правильных распознаваний для экспериментов на стенде «Мехатроника» при использовании предлагаемой математической модели СНС и в скобках без неё

Объект/Расстояние от камеры	50 см	1 м	1.5 м (расстояние до стола)	2 м

Мяч	93% (80%)	91% (75%)	87% (66%)	78% (63%)
Мотоцикл	91% (85%)	87% (79%)	88% (74%)	75% (69%)
Машина	89% (81%)	85% (77%)	82% (71%)	68% (67%)
Фонарь	94% (82%)	90% (71%)	92% (65%)	72% (59%)
Лягушка	95% (79%)	91% (78%)	89% (60%)	69% (59%)
Самолёт	91% (84%)	90% (72%)	87% (67%)	74% (65%)
Плеер	90% (79%)	89% (77%)	90% (63%)	75% (58%)
Тыква	85% (81%)	83% (71%)	80% (68%)	63% (63%)
Солдат	96% (85%)	92% (76%)	90% (64%)	73% (60%)
Степлер	92% (78%)	89% (64%)	88% (62%)	71% (59%)
Пустой класс	98%	98%	98%	96%

Из полученных результатов следует, что точность распознавания для объектов достигает 96% при использовании метода синтеза параметров математической модели СНС с расширенным обучающим множеством. С увеличением расстояния от камеры точность распознавания падает из-за ухудшающегося качества входного изображения. На расстоянии больше 2 м происходит выделение слишком большого количества потенциально интересных мест и распознавание в реальном времени становится невозможным. Обученная СНС может быть перенесена на мобильного робота AR-601E, где способна стать ядром системы его технического зрения.

### Список литературы

1. Галушкин А.И. Теория нейронных сетей : учеб. пособие для вузов / общая ред. А.И. Галушкина. – М. : ИПРЖР, 2000. – Кн. 1. - 416 с.
2. Горбань А.Н., Россиев Д.А. Нейронные сети на персональном компьютере / Рос. акад. наук. – Красноярск, 1996. – 270 с.
3. Лагунов Н.А., Мезенцева О.С. Влияние предобработки изображений на качество обучения нейронной сети для их распознавания // Вестник Северо-Кавказского государственного технического университета. – 2015. – № 1 (46). – С. 21–25.
4. Немков Р.М., Мезенцева О.С. Динамическое изменение воспринимающих свойств сверточных нейронных сетей как обучение с шумом и его влияние на обобщающую способность // Нейрокомпьютеры: разработка, применение. – 2015. – № 2 (63). – С. 12–18.
5. Хайкин С. Нейронные сети. Полный курс. - 2-е изд. / пер. с англ. – М. : Вильямс, 2008. – 1104 с. : ил.

6. Антропоморфный робот AR-600E. - URL: <http://npo-at.com/products/ar-600e/> (дата обращения: 08.05.2015).
7. Hassoun M. Fundamentals of Artificial Neural Networks. – A Bradford Book, 2003. – 511 с.
8. LeCun Y., Bottou L. Gradient-based Learning Applied to Document Recognition / Y. LeCun, L. Bottou, Y. Bengio, P. Haffner // Proceedings of the IEEE. - 1998. - V. 86. - P. 2278-2324.
9. Nemkov R., Mezentseva O. The Use of Convolutional Neural Networks with Non-specific Receptive Fields // The 4<sup>th</sup> International Scientific Conference: Applied Natural Sciences. – Nový Smokovec, 2013. – P. 284–289.
10. Modified National Institute of Standards and Technology (MNIST). - URL: <http://yann.lecun.com/exdb/mnist/> (дата обращения: 08.05.2015).
11. NYU Object Recognition Benchmark (NORB). - URL: [www.cs.nyu.edu/~ylclab/data/norb-v1.0/](http://www.cs.nyu.edu/~ylclab/data/norb-v1.0/) (дата обращения: 08.05.2015).

**Рецензенты:**

Лубенцов В.Ф., д.т.н., профессор, заместитель директора по научной работе Невинномысского технологического института (филиал) ФГАОУ ВПО «Северо-Кавказский федеральный университет» Минобрнауки РФ, г. Невинномысск;

Дроздова В.И., д.ф.-м.н., профессор, заведующая кафедрой информационных систем и технологий Института информационных систем и телекоммуникаций ФГАОУ ВПО «Северо-Кавказский федеральный университет» Минобрнауки РФ, г. Ставрополь.