

Image Recognition by a Second-Order Convolutional Neural Network with Dynamic Receptive Fields

Roman Nemkov
nemkov.roman@yandex.ru

Oksana Mezentseva
omezentceva@ncfu.ru
Maksim Brodnikov
brodmv@gmail.com

Dmitriy Mezentsev
terrakoto@me.com

Department of Information Systems & Technologies, North-Caucasus Federal University,
2, Kulakov Prospect, Stavropol, Russian Federation

Abstract

This paper presents the method description of convolutional neural networks parameters synthesis with dynamic receptive fields and second-order neurons for convolutional layers. The experiments of pattern recognition which shown that the combination of second-order neurons and dynamic receptive fields allows to reduce the generalization error are described in the article.

1 Introduction

Today, convolutional neural networks (CNN) are widely used for the invariant image recognition. This is due to the fact that with the help of this type of artificial neural networks (ANN) one can produce the better results on average 10% -15% higher than with conventional ANNs [1].

To date, there are various modifications of the CNNs [2]. However, the general trend remains the same: the increase of layers quantity (for the increase of non-linearity), the increase of different types layers (for high-quality processing of the input information, and feature extraction). If quantity of layers increases than computational costs increase too, so the better way is the creation of new types of CNN layers.

The author's modification of CNN on the basis of the dynamic receptive fields (RF) use instead of the classical square stationary RFs was described in [3, 4, 5]. CNNs with such fields can expand an input training set by the distortion of the primary patterns on different convolution layers. As a result of this process the error of generalization and learning is reduced.

In this paper, the mathematical model of CNN with dynamic RFs was changed by means of second-order neurons. It led to an even greater decrease of generalization error.

2 Adaptation of a Convolution Neural Network with Dynamic Fields towards Second-Order Neurons

CNN idea with dynamic RFs is that if one changes the set of RFs for some layers then the same pattern can be perceived in different ways by the network. Hence, one can extend a training set. It is known that the classical form of a RF for a CNN is a square. It is offered to use a template for non-standard RF generation. The elements of a template are indices indicating neighbors within two discrete steps from the element on a pixel array. If you change all RFs lying on a feature map, additional information will impact on the configurable parameters. It will lead to the extract of a better invariant (Fig. 1).

Copyright © 2017 by the paper's authors. Copying permitted for private and academic purposes.

In: S. Hölldobler, A. Malikov, C. Wernhard (eds.): *YSIP2 – Proceedings of the Second Young Scientist's International Workshop on Trends in Information Processing, Dombai, Russian Federation, May 16–20, 2017*, published at <http://ceur-ws.org>.

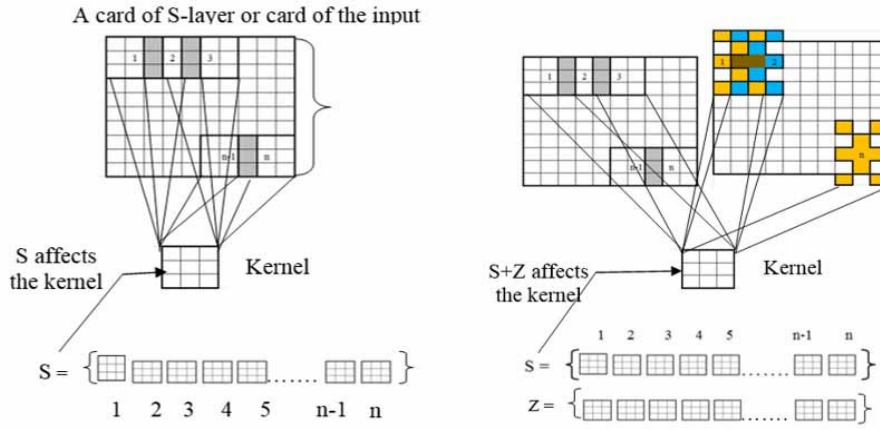


Figure 1: Left: standard RFs, Right: standard and addition RFs

High-order CNNs are expansion over signal direct distribution networks where polynomial neurons summators are used. That type of neurons enables a network to extract information from the input signal more qualitatively [6]. At the same time high-order functions (square, cube, trigonometric, etc.) are used instead of the usual weighted summation. The main drawback of this approach is the need to introduce additional coefficient arrays, but it is offset by higher quality features extracting (hence the generalization error reduces).

The activation function of the standard neuron with two inputs is calculated according to the formula 1, the activation function of the second-order neuron is calculated according to the formula 2.

$$f(w_1 * x_1 + w_2 * x_2) = f\left(\sum_{i=1}^2 w_i * x_i\right) \quad (1)$$

$$f(w_1 * x_1 + w_2 * x_2 + u_1 * x_1 * x_1 + u_2 * x_2 * x_2 + u_3 * x_1 * x_1) = f\left(\sum_{i,j} x_i * w_i + u_j * x_j^2\right) \quad (2)$$

where \mathbf{w} , \mathbf{u} are vectors of adjustable coefficients, and \mathbf{x} is an input vector, $f(\cdot)$ is the activation function of a neuron.

This approach does not touch activation functions of neurons and the principles of error calculation, so the only difference in contrast to the standard back-propagation algorithm is the calculation of the weighted sum.

We need to calculate two partial derivatives according to the formula 3 with the aim to know how much we want to change the weights \mathbf{w} and \mathbf{u} .

$$\begin{cases} \frac{\partial v_j(n)}{\partial w_{ji}(n)} = (\sum_{i=0}^m w_{ji}(n) * y_i(n) + u_{ji}(n) * y_i^2(n))' |_{w_{ji}(n)} = \sum_{i=0}^m y_i(n) \\ \frac{\partial v_j(n)}{\partial u_{ji}(n)} = (\sum_{i=0}^m w_{ji}(n) * y_i(n) + u_{ji}(n) * y_i^2(n))' |_{u_{ji}(n)} = \sum_{i=0}^m y_i^2(n) \end{cases} \quad (3)$$

where $v_j(n)$ is a weighted sum of the neuron j at iteration n , $y_i(n)$ is an input values for the neuron j .

The algorithm [7] which changes RFs for neurons belonging to any combination of convolutional layers (C-layers) before feeding a pattern was developed. The synthesis method of mathematical model parameters of CNN with dynamic RFs and second-order neurons for C-layers is proposed. This method was developed on the basis of the change algorithm of RFs and adaptation of the forward and back-propagation to neurons of second-order (2, 3). This method adapts the back-propagation algorithm, it consists of the following steps:

1. In the process of training before entering the next pattern, we need to change neurons RFs at the desired combination of C-layers using the algorithm of RFs form changing.
2. During the forward propagation we need to calculate the output values of C-layer neurons according to the

formula 4.

$$\begin{cases} C_{m,n}^i = \varphi(b + \sum_{q \in Q_i} \sum_{k=0}^{K_C-1} \sum_{l=0}^{K_C-1} ([X_{m+k+F_i(\cdot),n+l+F_j(\cdot)}^q * W_{k,l}^q] + [X_{m+k+F_i(\cdot),n+l+F_j(\cdot)}^q * A_{k,l}^q])) \\ S_{m,n}^i = \varphi(b + u \sum_{k=0}^{K_S-1} \sum_{l=0}^{K_S-1} C_{m*K_S+k,n*K_S+l}^i) \end{cases} \quad (4)$$

where $C_{m,n}^i$ is the neuron output disposed at i-th map of C-layer in position m, n, $\varphi(\cdot) = A * \tanh(B * p)$ when $A = 1.7159$, $B = 2/3$, b is a bias, Q_i is indices set of the previous layer cards associated with C^i card, K_C is the size of square RF for the neuron $C_{m,n}^i$, $X_{m+k,n+l}^q$ is an input value to the neuron $C_{m,n}^i$, the vectors \mathbf{W} and \mathbf{A} are custom weights for the second-order neurons of C-layer, $S_{m,n}^i$ is the neuron output of averaging layer; $F_i(\cdot)$ and $F_j(\cdot)$ are the $F_i(RF_{m,n}, k, l)$, $F_j(RF_{m,n}, k, l)$ (Functions which return the offsets for the row and the column for RF-template belonging to neuron m, n at position k, l within the template). $index_{k,l}$ is a template element of at position k, l, $index_{k,l} = 0..24$.

These functions are determined by the following formulas:

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

3. During the back propagation we need to receive the local gradient for C-layer according to the formula 6.

$$\begin{cases} \frac{\partial E}{\partial (W_{k,l}^\lambda)^q} = \sum_{m=0}^{SizeC} \sum_{n=0}^{SizeC} \delta_{m,n}^\lambda * y_{m+k+F_i(RF_{m,n}^\lambda, k, l), n+l+F_j(RF_{m,n}^\lambda, k, l)}^{\lambda-1} \\ \frac{\partial E}{\partial (A_{k,l}^\lambda)^q} = \sum_{m=0}^{SizeC} \sum_{n=0}^{SizeC} \delta_{m,n}^\lambda * y_{m+k+F_i(RF_{m,n}^\lambda, k, l), n+l+F_j(RF_{m,n}^\lambda, k, l)}^{\lambda-1} \end{cases} \quad (6)$$

where $\delta_{m,n}^i$ is a residual was gathered for a neuron with coordinates m, n within the map of layer λ , q is the part of the kernel of configurable features for which we receive the components of the gradient, SizeC is the size of the C-layer card, $A_{k,l}^q$, $W_{k,l}^q$ is q-th part of the adjustable parameters, which is responsible for interaction with the q-th card of the previous layer.

CNN learning algorithm was developed on the basis of the proposed adaptation method of back propagation algorithm to second-order neurons and dynamic RFs (Fig. 2).

3 Experiments

Experiments were carried out to evaluate the generalization capability of a CNN with second-order neurons and dynamic RFs.

Stereo dataset Small NORB [8] was used as a sample database for the invariant pattern recognition. The use of classical CNN of LeNet-5 type gives the generalization error of 8.4% [9].

The use of CNN with dynamic RFs and the general structure of LeNet-5 gives the generalization error of 4.3% [10]. Table 1 shows the results of LeNet-5 with dynamic RFs and second-order neurons in various combinations of the C-layer.

Where C_i , S_i in the Table 1 are the numbers of corresponding convolution and averaging layers, I is the input layer, O is the output layer, number 2 means the second-order neurons in the corresponding layer. The value of the generalization error was calculated as the average of 15 experiments.

The table shows that not all combinations of the second-order neurons used in the C-layers provide a significant reduction of the generalization error. The best results show the embodiment number 6 and number 9. The best

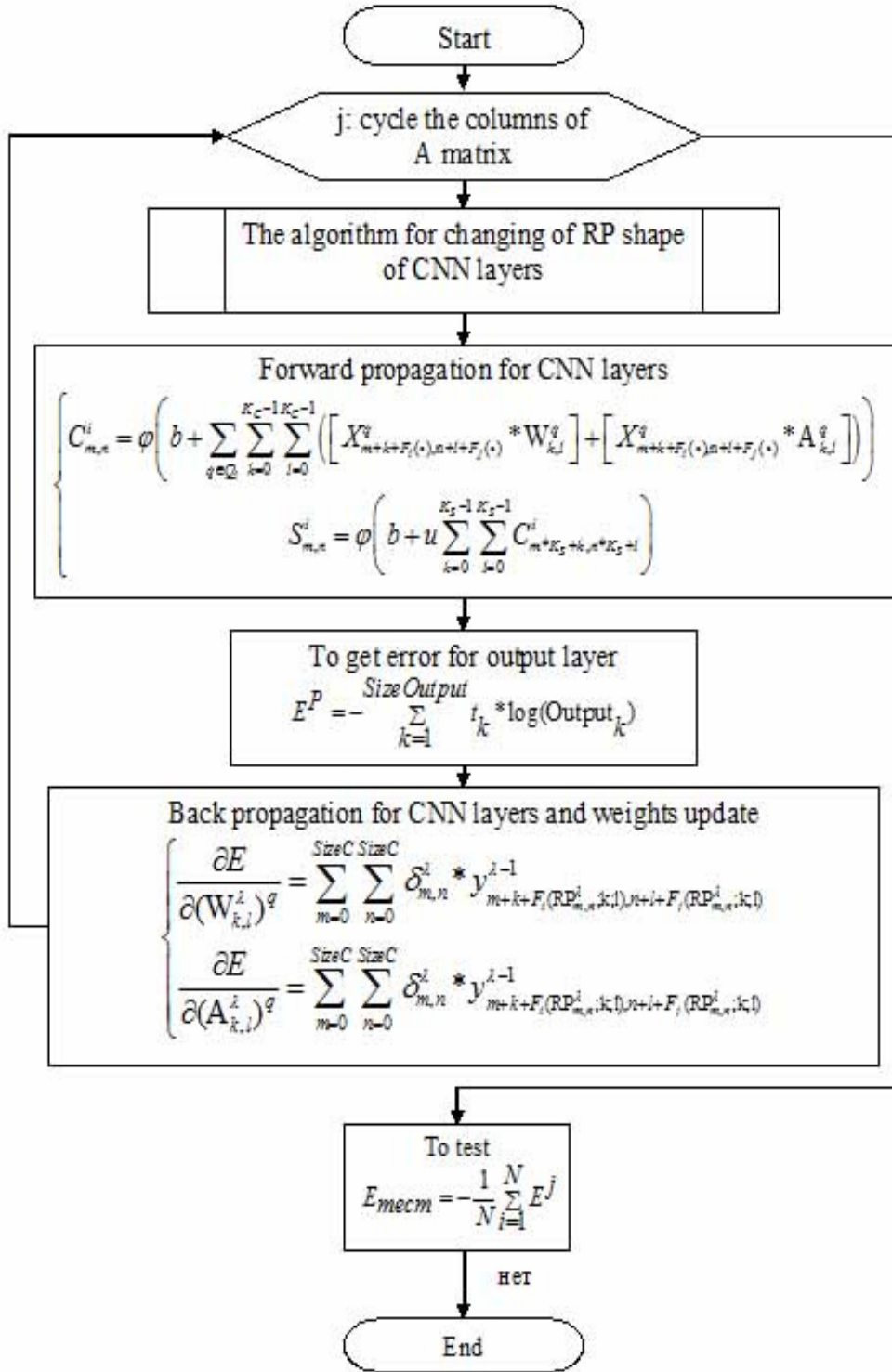


Figure 2: The CNN learning algorithm with dynamic RFs, and second-order neurons

Table 1: Results of the tests for different variants of second-order neurons using

Number	I	C1	S1	C2	S2	C3	O	Error
1								4.3%
2							2	4.4%
3		2						4.1%
4				2				4.2%
5						2		4.2%
6		2		2				2.5%
7				2		2		3.6%
8		2				2		3.8%
9		2		2		2		2.3%

combination can only be found through exhaustive search of possible values. It limits the use of higher order CNN.

As a result of the proposed CNN with dynamic RFs and second-order neurons we were able to reduce the generalization error by an average of 2% in comparison with the use of the CNN with simply dynamic RFs and by 6.1% in comparison with the CNN with alternating C and S - layers. It is shown that the use of the artificial neural network with the higher-order neurons leads to the decrease of the generalization error. But as a result we have to carry out additional experiments in order to find the necessary combination of C-layers with the modified neural activation functions.

References

- [1] Goodfellow, I., Bengio, Y.: Deep Learning (Adaptive Computation and Machine Learning series). Adaptive Computation and Machine Learning series. 2016. P. 800.
- [2] Mirza, M., Courville, A., Bengio, Y.: Generalizable Features From Unsupervised Learning, 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 4945-4949, 2016
- [3] Nemkov, R., Mezentseva, O., Mezintsev, D.: Using of a Convolutional Neural Network with Changing Receptive Fields in the Tasks of Image Recognition / Proceedings of the First International Scientific Conference Intelligent Information Technologies for Industry (IITI16), Volume 451 of the series Advances in Intelligent Systems and Computing, 2016, pp. 15-23.
- [4] Nemkov, R., Mezentseva O.: The Use of Convolutional Neural Networks with Non-specific Receptive Fields. The 4th International Scientific Conference: Applied Natural Sciences. - Novy Smokovec, 2013. - pp. 284 - 289.
- [5] Nemkov, R. M., Mezentseva O. S.: Dynamical change of the perceiving properties of convolutional neural networks and its impact on generalization. Neurocomputers: development and application, 2015, no. 2, pp. 12-18.
- [6] Lagunov, N.: Allocation and recognition of objects with the help of optimized algorithm of selective search and a high-order convolutional neural network // Fundamental research. 2015. 5. pp. 511-516.
- [7] Nemkov, R.: The method of a mathematical model parameters synthesis for a convolution neural network with an expanded training set. The modern problems science and education, 2015. 1. URL: <http://www.science-education.ru/125-19867/>;
- [8] NYU Object Recognition Benchmark (NORB). URL: www.cs.nyu.edu/~ylclab/data/norb-v1.0/ (28.02.2017).
- [9] Nemkov, R.: Dynamical Change of the Perceiving Properties of Neural Networks as Training with Noise and Its Impact on Pattern Recognition / R. Nemkov // Young Scientists International Workshop on Trends in Information Processing (YSIP) 2014. URL: <http://ceur-ws.org/Vol-1145/paper4.pdf/>
- [10] Nemkov, R. M.: Synthesis method of mathematical model parameters of the convolutional neural network with extended training set. URL: <http://www.science-education.ru/125-19867/> (30.01.2016).