ESANN 2021 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 6-8 October 2021, i6doc.com publ., ISBN 978287587082-7.
Available from http://www.i6doc.com/en/.

Correlated Weights Neural Layer with external control

Slawomir Golak *

Department of Industrial Informatics
Silesian University of Technology, Gliwice, Poland

Abstract. The correlated weights neural layer is a generalization of the convolutional layer constituting the core of CNN networks. The CWNL layer takes advantage of weights correlated with coordinates of a neuron and its inputs, calculated by a dedicated neural subnet. In this work, a modified CWNL layer is proposed, which allows the parameterized spatial manipulation (and any other global transformation) of a pattern. The externally controlled CWNL layer can be used in existing neural network architectures, giving them the ability of internal pattern transformation without any modification of the training process.

1 Introduction

Convolutional neural networks (CNNs) are effective models which recently enjoyed great success e.g. in image and video processing. The main benefit of using CNNs is the reduced amount of parameters that have to be determined during a learning process. The CNN can be regarded as a variant of the standard neural network which instead of using fully connected hidden layers, introduces a special network structure, which consists of alternating convolution and pooling layers. The layer performs the convolution operation using data from the weight matrix - the kernel. A single weight in the convolutional layer (CNL) can be defined as a function of the relative position of a neuron and its input:

$$w = f(\boldsymbol{P}^{(I)} - \boldsymbol{P}^{(O)}) \tag{1}$$

where: $\boldsymbol{P}^{(O)}$ - vector of a neuron spatial coordinates, $\boldsymbol{P}^{(I)}$ - vector of spatial coordinates of a neuron input. The function $f()$ itself is actually a kernel matrix indexed by difference $\boldsymbol{P}^{(I)} - \boldsymbol{P}^{(O)}$. Its parameters are discontinuous and limited by the kernel size. There have been some attempts to use more complex definitions of the weight function. For example, the deformable convolutional networks [3] allow to extend the spatial range of feature detection. Basing weights on relative coordinates causes a loss of information about the absolute position of a neuron and its inputs and prevents the implementation of the most popular global pattern transformations (e.g. rotation, scaling, reflection). The solution proposed in papers [1, 2] is to describe a weight of a connection between a neuron and its input by a function using absolute spatial coordinates of the neuron and

*This research was supported by grants of Silesian University of Technology (11/040/RGJ20/0017 and 11/040/BK21/0023) and by PLGrid Infrastructure.

ESANN 2021 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 6-8 October 2021, i6doc.com publ., ISBN 978287587082-7.
Available from http://www.i6doc.com/en/.

its input. The cost of this solution is the necessity to replace the implementation of the function by the matrix indexed with the relative coordinates of an input and a neuron by some universal approximator. The papers propose the possibility of using a neural subnet for this task, which in the learning process determines the correlation between the spatial, absolute coordinates of the neuron, its input, and the weight value. In both the CNL layer and the original version of the CWNL layer, we are dealing with a static transformation, the course of which cannot depend on additional external signals. In this paper, it is proposed to modify the CWNL layer by extending the definition of the weight function and including additional parameters, external to the layer, controlling the work of the layer:

$$w = f(\boldsymbol{P}^{(I)}, \boldsymbol{P}^{(O)}, \boldsymbol{S}^{(E)}) \tag{2}$$

where: $\boldsymbol{S}^{(E)}$-vector of external signals. One of the possible applications of such a layer and the inspiration for starting work on its development was the emergence of the concept of contextual networks [4]. The structure of these networks includes modules whose task is to transform (in the main processing path) the pattern signal based on parameters determined in the processing path parallel to the main one. This solution allows the neural network to eliminate geometrical deformations of the classified pattern and is a new technique for creating transform-invariant classifiers. In the cited paper the operation itself is carried out with the use of a mapping grid which has a limited range of possible transformations.
A neural layer whose transformation can be externally controlled is the type of layer used in steerable CNNs [5]. An important limitation of this solution is basing it on a predefined (at the design stage) of types of transformations.
The CWNL layer with external control is a much universal component for the parameterized spatial transformation, which should at least theoretically acquire the ability to perform any transformation during the learning process using very small training sets.

2 Model of the CWNL layer with external control

The structure of the correlated weights neural layer with external control is shown in Fig. 1. The position of each layer input is described by the spatial coordinates vector $\boldsymbol{P}^{(I)}$, which size is compatible with the dimensionality of the input data. The topology of the CWNL layer is compatible with dimensionality and the size of its output pattern. Each of its neurons also has its own spatial, absolute coordinates $\boldsymbol{P}^{(O)}$. The output of the CWNL neuron is determined based on the standard equation (3). The difference is that in the equation, instead of the constant weights and the bias, their values are calculated dynamically by dedicated neural subnets.

$$y_i^{(O)} = f\left(s_i\right) \quad \& \quad s_i = \sum_{j=1}^{N^{(O)}} \omega_{ij0}\left(\boldsymbol{P}_i^{(O)}, \boldsymbol{P}_j^{(I)}, \boldsymbol{S}^{(E)}\right) y_j^{(I)} + \beta_{i0}\left(\boldsymbol{P}_i^{(O)}, \boldsymbol{S}^{(E)}\right) \tag{3}$$

ESANN 2021 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 6-8 October 2021, i6doc.com publ., ISBN 978287587082-7.
Available from http://www.i6doc.com/en/.

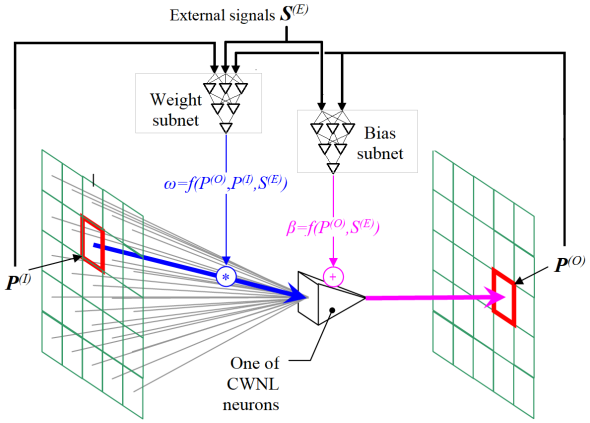

Fig. 1: Schematic diagram of a correlated weights neural layer with external control.

where: $f()$ - activation function of the CWNL layer, $y_i^{(O)}$ - output of i-th neuron in the layer (output of the layer), $y_j^{(I)}$ - j-th input of the layer, $\omega_{ij0}()$ - dynamically calculated weight, $\beta_{i0}()$ - dynamically calculated bias, $\boldsymbol{P}_i^{(O)}$ - vector of i-th neuron spatial coordinates, $\boldsymbol{P}_j^{(I)}$ - vector of spatial coordinates of j-th layer input, $\boldsymbol{S}^{(E)}$ - vector of external signals. The function $\omega()$ that dynamically calculates the value of weights based on the coordinates of the neuron and the input is implemented by a neural subnetwork with standard fully connected layers:

$$\sigma_{ijk}^{(\omega,m)} = \sum_{l=1}^{N^{(\omega,m-1)}} w_{kl}^{(\omega,m)} \omega_{ijl}^{(m-1)} + b_k^{(\omega,m)} \tag{4}$$

$$\omega_{ijk}^{(m)} = \phi^{(\omega,m)}\left(\sigma_{ijk}^{(\omega,m)}\right) \quad \& \quad \omega_{ijk}^{(0)} = \{\boldsymbol{P_i}^{(O)} \cup \boldsymbol{P_j}^{(I)} \cup \boldsymbol{S}^{(E)}\}_k \tag{5}$$

where: $\omega_{ijk}^{(m)}$- processed by the subnetwork signal of the dynamically calculated weight for i-th neuron and j-th input (of the CWNL layer) in m-th subnet layer, $\phi^{(m)}()$ - activation function of m-th subnetwork layer, $w_k l^{(\omega,m)}$- weight of l-th input of k-th neuron, $b_k^{(\omega,m)}$ - bias of k-th neuron. The input of the subnetwork calculating weights $\omega_{ijk}^{(0)}$ is the union of the vector of the neuron coordinates $\boldsymbol{P}^{(O)}$, the vector of the layer input coordinates $\boldsymbol{P}^{(I)}$ and the vector of external control signals $\boldsymbol{S}^{(E)}$.

A bias in a neural layer can be defined as a weight of an input with a constant value of 1. In the case of a convolution layer, for one feature map, a single value of the bias is repeatedly used for each neuron of this layer. The layer CWNL allows to dynamically calculate the bias value based on the coordinates of the neuron supplemented with external control signals. This task is performed by

ESANN 2021 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 6-8 October 2021, i6doc.com publ., ISBN 978287587082-7.
Available from http://www.i6doc.com/en/.

the second dedicated subnet. Since each neuron has only one bias value, the subnet does not use the coordinates of the inputs:

$$\sigma_{ik}^{(\beta,m)} = \sum_{l=1}^{N^{(\beta,m-1)}} w_{kl}^{(\beta,m)} \beta_{il}^{(m-1)} + b_k^{(\beta,m)} \tag{6}$$

$$\beta_{ik}^{(0)} = \{\boldsymbol{P_i}^{(O)} \cup \boldsymbol{S}^{(E)}\}_k \quad \& \quad \beta_{ik}^{(m)} = \phi^{(\beta,m)}\left(\sigma_{ik}^{(\beta,m)}\right) \tag{7}$$

If activation functions of the CWNL layer and subnets layers are continuous and differentiable, a learning process can be carried out based on any gradient optimization technique [2].

3 Experiment

The experiment was to test the ability of the network with the CWNL layer to carry out a multi-parameter affine transformation of images. The transformation involved simultaneous translation, rotation and scaling. The research was based on images of digits from the MNIST set. The input was the original digit image and transformation parameters (standardized), the output was the transformed image. To implement the above-mentioned transformation, a neural network with a single CWNL layer was used. The layer used a sigmoidal activation function due to the range of values of the returned pixels of the images (0-1). The structure of the subnet generating weight values was *I4-H32-H16-H8-O1*. The subnet calculating biases had layers *I4-H8-H4-O1*. The ELU activation function of hidden layers prevents the vanishing gradient problem better than the popular ReLU function and at the same time allows for a better representation of dependencies in data [6]. The learning process was based on the minimization of the binary entropy loss function. The classic RProp learning algorithm was used, which proved to be effective for the developed network architecture than the currently used algorithms. Figure 2 (the second section) shows the results obtained for the above-described network for the training set with the size of 96 cases. For such a small number of the training set, the obtained quality of the transformed images fully confirms the ability of the developed network to implement this type of transformation. The unusual size of the set results from the architecture of the computing cluster used and the way of parallelizing the calculations by splitting the training set.

For comparison, a network with fully connected layers was used to implement the same transformation. The structure of the network was modeled on the structures of autoencoders using FC layers (supplemented with additional inputs for control parameters), which confirmed their ability to process digits from the MNIST set. Used neural network structure consisted of 5 hidden layers with the ELU activation function and a sigmoid output layer, its layout: *I(28*28+4)-H128-H64-H32-H64-H128-O(28*28)*. The analysis of the obtained results (Fig. 2, the thrid section) allows us to conclude that the proposed network with FC layers can realize the expected affine transformation. However, an acceptable

ESANN 2021 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 6-8 October 2021, i6doc.com publ., ISBN 978287587082-7.
Available from http://www.i6doc.com/en/.

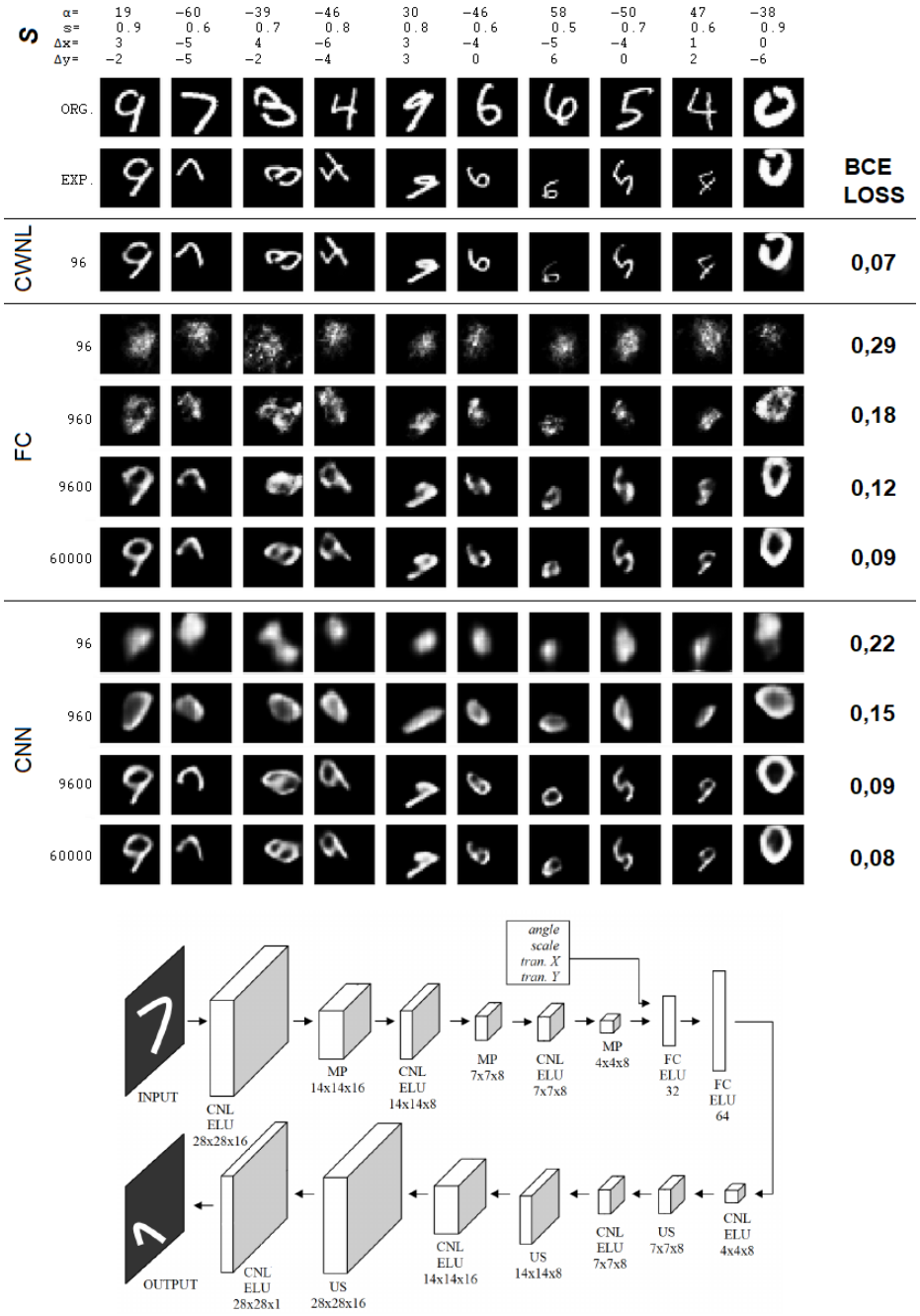

Fig. 2: Affine transformation using networks based on the CWNL layer, full connect layers and convolutional layers (show in the lower section): input image, input control parameters (angle, scale, translation x,y); expected images from testing set and images after learning on the training set containing 96-60000 examples with the binary cross-entropy loss.

ESANN 2021 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 6-8 October 2021, i6doc.com publ., ISBN 978287587082-7.
Available from http://www.i6doc.com/en/.

level of transformation quality is only achieved for a large training set. This means an increase in the required size of the training well over two orders of magnitude. Even in this case, the quality of the obtained images is much worse. Global error measures also show (BCE, in Fig. 2) the advantage of the network containing a single CWNL layer.
A network with CNL layers is better suited to image processing. However, a big problem is the introduction to them of external control signals, the topology of which is completely different from the topology of the processed patterns. A modified convolutional autoencoder structure was used for this task, in which external signals were delivered to the middle layers starting the block of the pattern decoder (Fig. 2, the lower section). Using the network configured in this way, slightly better results (Fig. 2) were obtained than the network using only FC layers, with the same, high requirements for the size of the training set.

4 Conclusions

The conducted research has shown that networks using the CWNL layer can effectively implement parameterized image transformations (and by generalizing to any patterns with the spatial distribution of the data). Due to the relatively small number of parameters describing such a layer, not related to the size of the pattern, but the complexity of its content, small data sets are enough to train a network with the CWNL layer. It can be assumed that with more complex transformations the requirements for the size of the training set will increase, but this size will be at least an order of magnitude smaller than in the case of existing architectures. Of course, in real applications, the CWNL layer controlled by external signals will not be a standalone component of the network, but one of its many elements. However, its special properties give the chance to obtain better results in the classical problems of classification, generation of artificial patterns, compression, and possibly in applications that will be specific to networks using this new layer type.

References

[1] S. Golak, Induced weights artificial neural network. In proceedings of the*International Conference on Artificial Neural Networks* (ICANN 2005), LNCS 3697, pages 295-300, Springer Berlin Heidelberg 2005.

[2] S. Golak, et al., New architecture of correlated weights neural network for global image transformations, In proceedings of the *International Conference on Artificial Neural Networks* (ICANN 2018), LCNS 11140, pages 56-65, Springer 2018.

[3] X. Zhu, et al., Deformable convnets v2: More deformable, better results, In proceedings of the *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9308-9316, Long Beach, CA, USA, 2019.

[4] M. Jaderberg, et al, Spatial transformer networks. In procedings of *Conference on Neural Information Processing Systems* (NIPS 2015), pages 2017-2025, Montreal, Canada, 2015.

[5] T. S. Cohen, M. Welling, Steerable CNNs. In procedings of the *International Conference on Learning Representations*, (ICLR), 2017.

[6] A. Ciuparu, et al., Soft++, a multi-parametric non-saturating non-linearity that improves convergence in deep neural architectures, *Neurocomputing*, 384:376-388, Elsevier 2020.