

# Convolutional Neural Network Feature Reduction using Wavelet Transform

A. Levinskis<sup>1</sup>

<sup>1</sup>*Department of Transport Electronics and Telematics, Riga Technical University, Lomonosova St. 1, V block, LV-1019, Riga, Latvia, phone: +371 27554968  
aleksandrs.levinskis@rtu.lv*

**Abstract**—Paper describes wavelet transform possible application for convolutional neural networks (CNN). As it already known, wavelet transform gives good signal representation in time and frequency domains. This can be useful for CNN input feature reduction as well as architecture simplicity by using only part of coefficients. The result of work is set of experiment which enables to configure out the most appropriate coefficient part. After feature reductions and architecture simplicity achieved configuration could classify data almost ten times faster than original.

**Index Terms**—Wavelet transform, artificial, neural networks, frequency estimation.

## I. INTRODUCTION

Artificial neural networks have found application in wide range of modern technologies such as image and voice recognition, natural language processing, speech synthesis, ect. However, neural network model of multilayer perceptron (MLP) allows to decide majority of classification problems it has some drawbacks. First of all it doesn't take into account a two dimensional spatiality of input data [1]–[7]. This is main disadvantage for classifying images and other 2D data. The decision of this problem comes from biology visual cortex model where neurons are grouped into receptor fields. This allows analyze data spatially and define features of it. The convolutional neural network represents such model [1].

Convolutional neural networks (CNN) have different architecture form MLPs, which definitely complicate learning process [3], [6]. The learning process directly depends on features count, which for image is its height multiplied by width in pixels. The author offers apply 2D discrete wavelet transform (DWT) for input data, therefore reducing number of features with saving performance of CNN. Next sections will describe CNN architecture and its properties.

## II. CNN ARCHITECTURE AND PROPERTIES

CNN consist of multiple convolution and sub sampling layers, where convolution and subsampling layers change each other. Each of the layers includes several features maps, which are connected to previous layer maps through

set of small receptive fields. In general CNN architecture is shown in Fig. 1.

As it can be expected input data is splitted into several features maps. It is performed by using different 2D kernels and convolution operation. The result of convolution defines features of input data, such as corners, curves, lines ect. Feature occurrence is mirrored on feature map by corresponding receptive field location, defining features map. After each convolution layer follows subsampling layer. It performs data reduction for future analysis by reducing each map dimension in four times.

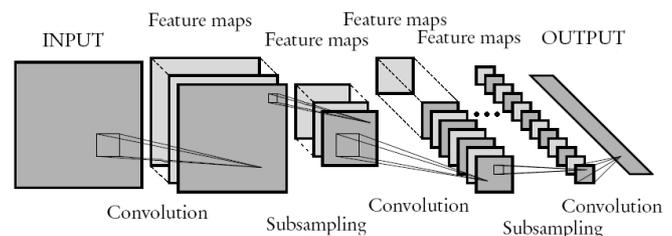


Fig. 1. CNN architecture.

Every subsampling layer is followed by convolution layer. In this case connections between layers defines interconnection matrix.

This process repeats until features maps become too small, e.g. 1x1. After this feature maps is followed by full interconnected MLP which output is classifiers vector.

The idea of performing subsampling layer is shown on Fig. 2.

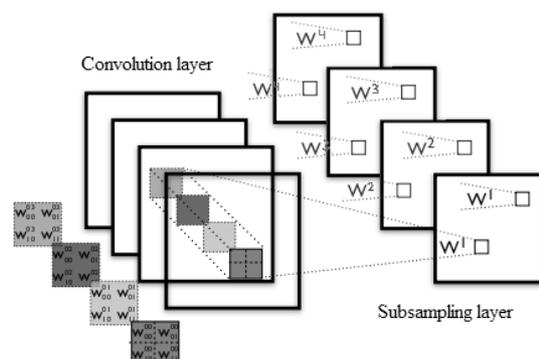


Fig. 2. Subsampling layer formation.

Hence  $w^1$  and  $w^2$  is 2D kernels, which coefficients in this case are  $w_{00}^{01}, w_{01}^{01}, w_{10}^{01}, w_{11}^{01}$  and  $w_{00}^{02}, w_{01}^{02}, w_{10}^{02}, w_{11}^{02}$  respectively. These kernels coefficients are defined during



samples. The learning plot and performance is shown at Fig. 3.

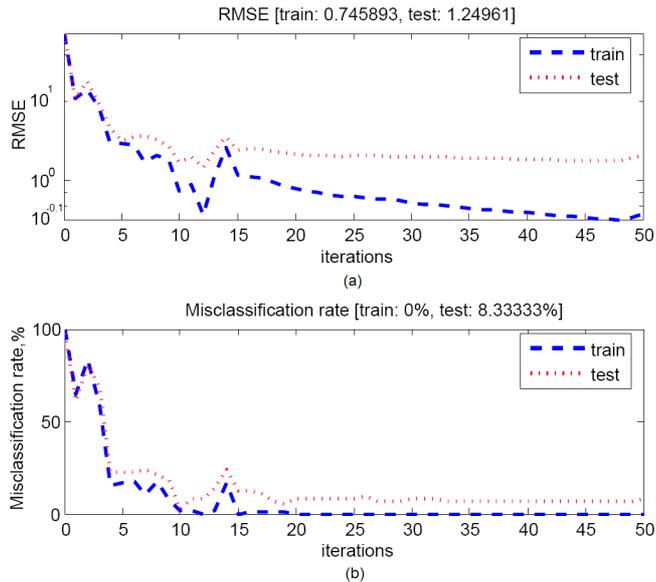


Fig. 3. Performance plots for usual training set: RMSE (a); misclassification rate (b).

As it can be expected Fig. 3(a) Root Mean Square Error (RMSE) for this case is 1.25 Fig. 3(b) and misclassification rate is 8.33% of test set. These parameters are defined after learning curve converges. RMSE corresponds to output classifier vector, but misclassification rate shows wrong classified symbols rate from test set.

For such CNN configuration data forward propagation time for each symbol is about 250 ms, but learning process takes 1 h 55 min. These times are used for reference; however they may depend on computation abilities. After reference computation is performed, the set of experiments is made.

For first experiment, (5) is applied for each of symbols. After performing transform, CNN is trained by such samples. How it can be noticed in Fig. 4(b), performance becomes even worse. This can be described by sample feature redundancy, what makes CNN unstable. In this case learning time as well as forward propagation time doesn't changes.

It implies new approach for using transformed data. As it can be expected in (5) transformed data consist of four parts, which includes approximation and detail levels coefficients. It can be appropriate to use one of the parts for CNN input features, but it is necessary to change CNN architecture for such samples. If input data will be quarter of original samples, then input layer should acknowledge 14x14 data structure. Therefore CNN input layer size should be decreased in four times. As the result, CNN internal layer number is decreased as well. It can be described by features maps earlier degradation.

After performing a bunch of experiment it was evaluated, that the most suitable part for CNN input is approximation coefficient part A, which gives better performance than others. It can be described also by intuition of using smooth copy of original samples.

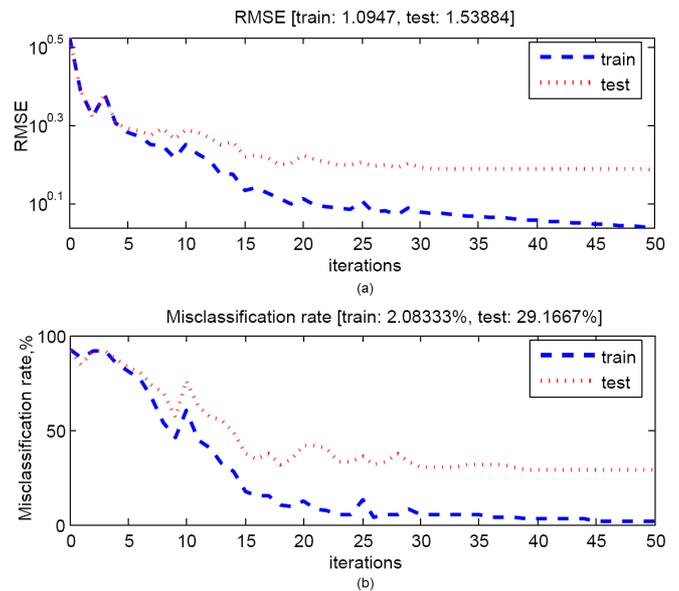


Fig. 4. Performance plot with (5) applied to samples: RMSE (a); misclassification rate (b).

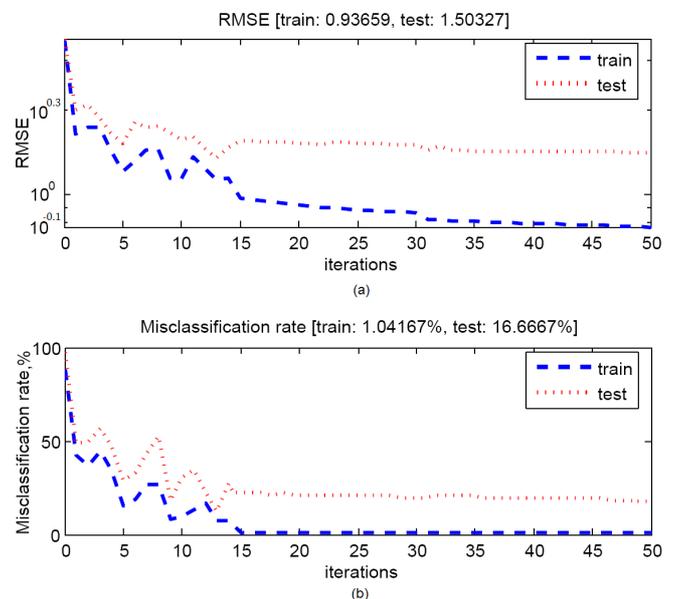


Fig. 5. Performance plot for approximation part of transformed original image: RMSE (a); misclassification rate (b).

For given CNN configuration learning time was 8 min 16 sec and data forward propagation time decreased to 27 ms, which is dramatically smaller compared to original LeNet5. But as it can be noticed test set RSME and misclassification rate is 1.5 and 16.7% respectively. This is limit for such architecture configuration, which is caused by CNN reduction and internal feature maps low count.

For getting better performance, the original LeNet5 first convolution layer maps count was extended from 6 up to 12. After learning process completed, it gives new performance plot, which is shown on next figure.

The last configuration of CNN architecture gives same misclassification rate as the first one, but learning time becomes 9 minutes instead 1 hour 55 min, as well data forward propagation time dropped from 250 ms to 32 ms. These results are quite exciting and shows suitable application of wavelet transform to CNN.

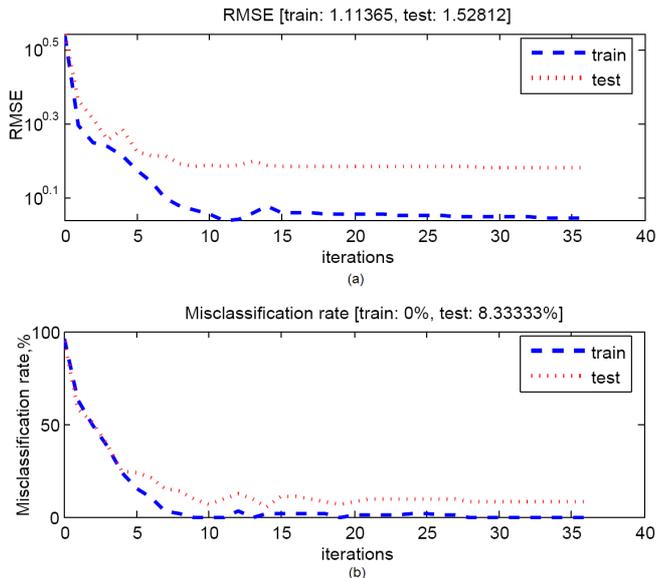


Fig. 6. Performance plot of modified LeNet5: RMSE (a); misclassification rate (b).

## V. CONCLUSIONS

The achieved results of this work shows advantage of applying wavelet transform for neural network feature reduction, which causes it architecture reduction. This results lower latency for data forward propagation as well as learning time becomes faster. In this particular example wavelet transform is applied to convolutional neural network. The next statements describe actual results of work:

- 1) Original LeNet5 learning time and data forward propagation latency was 1 h. 55 min and 250 ms respectively.
- 2) For modified LeNet5 architecture with applied wavelet transform to input data, learning time and data forward propagation latency becomes 9 min and 32 ms respectively.

Both configurations have same misclassification rate, what makes second one preferred.

## REFERENCES

- [1] В. Г. Прохоров. "Использование сверточных сетей для распознавания рукописных символов", *Проблемы программирования*, Институт программных систем НАН Украины, no. 2, pp. 669–774, 2008.
- [2] C. Christopoulos, A. Skodras, T. Ebrahimi, "The JPEG2000 still image coding system: an overview", *IEEE Transactions on Consumer Electronics*, pp. 1103–1127, 2000. [Online]. Available: <http://dx.doi.org/10.1109/30.920468>
- [3] Y. LeCun, L. Bottou, G. Orr, K. Muller, *Efficient BackProp in Neural Networks: Tricks of the trade*. Springer, 1998, p. 44.
- [4] S. Mallat, *A Wavelet Tour of Signal Processing*, 2<sup>nd</sup> ed., London, Academic Press, 1999, p. 637.
- [5] Y. L. Cun, L. Bottou, Y. Bengio, P. Haffner, "Gradient-based learning applied to document recognition", in *Proc. of the IEEE*, vol. 86, 1998, pp. 2278–2324. [Online]. Available: <http://dx.doi.org/10.1109/5.726791>
- [6] Y. L. Cun, B. Boser, J. Denker, D. Henderson, R. Howard, W. Hubbard, L. Jackel, "Backpropagation applied to handwritten zip code recognition", *Neural Computation*, pp. 541–551, 1989.
- [7] P. Y. Simard, D. Steinkraus, J. C., "Platt Best Practices for Convolutional Neural Networks Applied to Visual Document Analysis", in *Proc. of the International Conference on Document Analysis and Recognition*, 2003, pp. 958–962.