FAKULTA APLIKOVANÝCH VĚD ZÁPADOČESKÉ UNIVERZITY V PLZNI

STUDENTSKÁ VĚDECKÁ KONFERENCE 2017

Insight of Neural Network by Removing Synapses

Martin Bulín[1], Luboš Šmídl[2]

Introduction

Neural networks can be trained to work well for particular tasks, but hardly ever we know why they work so well. Due to the complicated architectures and an enormous number of parameters we usually have a well-working black box and it is hard if not impossible to make targeted changes in a trained model.

When we deal with a real (not academical) problem, one often comes to a point when his or her network works fine (let's say with an accuracy of 90%), but a customer asks for the accuracy of 98% for example. Then we can either keep trying and spend months on tuning the black box mostly in a random manner, or, if we demystify what is going on inside the network, we can suggest reasonable and targeted improvements.

My work is focused on understanding the behaviour of feedforward neural networks when classifying particular data. The method rests in removing unimportant synapses from a trained network, while the classification accuracy is kept. Based on my experience, over 90% of the synapses are usually redundant in fully-connected networks.

The hypothesis is that one can find some rules if a network consists of important parts only. This effort could possibly lead to a general knowledge of how to design networks and tailor them for challenged problems. Moreover, the dimensionality is significantly reduced, which speeds up learning and prediction, and could be useful for low-cost embedded systems.

Methods

Pruning methods (Reed, 1993) work with the hypothesis that some of the synapses in fully connected networks do not contribute to the classification and so their removal would not cause a significant accuracy drop.

Figure 1: Principle of pruning methods.

The key question is how to find the redundant synapses and distinguish them from important ones. My measure originates in a nice and simple idea: *The redundant synapses do not change their weights over the backpropagation-based training*.

$$WSF(w_k) = |w_k(t_f) - w_k(0)| \tag{1}$$

[1] master-degree student of Applied Sciences and Computer Engineering, field of study Cybernetics and Control Engineering, focused on Neural Networks, e-mail: bulinm@students.zcu.cz

[2] assistant professor at the Department of Cybernetics, e-mail: smidl@kky.zcu.cz

Hence, we get the `WSF` (*weight significance factor*) as a difference between the weight value after training $w_k(t_f)$ and its initial value $w_k(0)$. Those synapses with low `WSF` are considered less important than those with higher `WSF`.

Results

Figure 2 illustrates the pruning process on the well-known MNIST problem (LeCun et al., 2010). The original fully-connected network of structure `[784, 20, 10]` (15880 synapses) was reduced to structure `[465, 20, 10]` with 1259 synapses, while the classification accuracy of 97% was kept.

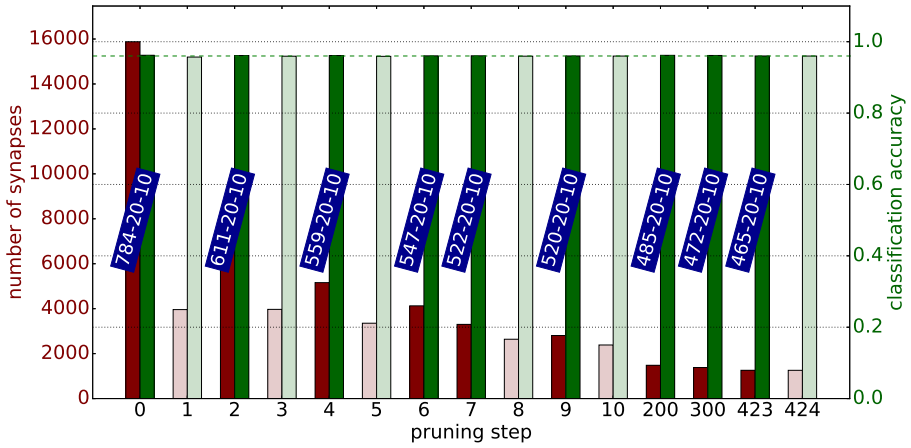

Figure 2: Pruning Process: problem of handwritten digits recognition.

Figure 3 shows a result of another experiment. The illustrated network is capable of learning the MNIST problem with the test accuracy of 50%, using 20 features and 38 synapses only. The colors distinguish synapses and features important for individual classes.

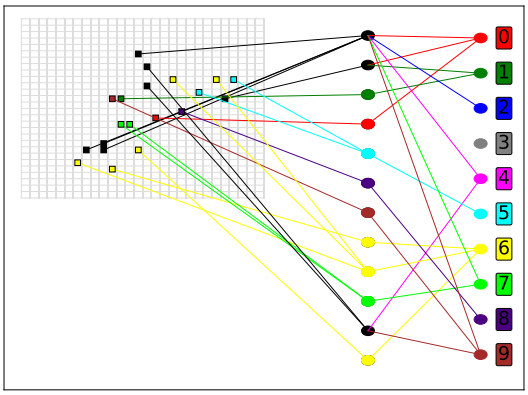

Figure 3: Demystification of individual parts in a pruned network.

References

Reed, R. (1993) Pruning algorithms - a survey. *IEEE transactions on Neural Networks 4.5.* pp. 740-747.

LeCun, Y. and Cortes, C. (2010) MNIST handwritten digit database. `http://yann.lecun.com/exdb/mnist/`