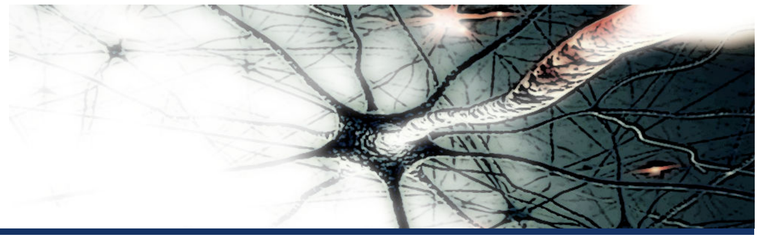
Optimization of Neural Network

Martin Bulín[1] and Luboš Šmídl[2]

Department of Cybernetics | Faculty of Applied Sciences | University of West Bohemia in Pilsen



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.

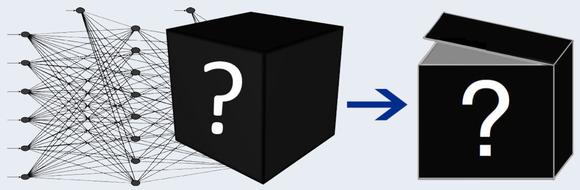


Figure: Let's open the black box to see what is going on inside.

In this thesis, we focus on network optimization, specifically we make networks small and simple by removing unimportant synapses, while keeping the classification accuracy of the original fully-connected networks.

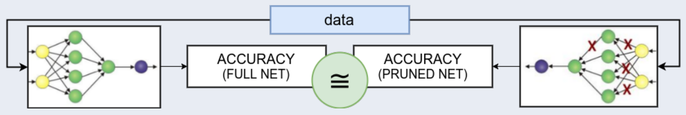


Figure: The principle of network pruning.

A pruned network then consists of important parts only and therefore we can find input-output rules and make statements about individual parts of the network. If we understand how a trained network actually works, we are simply able to make *targeted* changes to improve its classification performance.

Hypotheses:

1. Fully-connected networks have redundant synapses.
2. If only important synapses remain, we can find rules.

Methods and The Key Idea

The crucial question is: *"How do we distinguish the redundant synapses from the important ones?"* Here the novelty of the thesis comes: *"Weights of redundant synapses are not changed over a backpropagation-based training."* This simple idea leads to a definition of a new measure, called *weight significance factor*, computed for every single synapse k:

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

where $w_k(t_f)$ is a weight of synapse k after training and $w_k(0)$ is its initial value. We iteratively remove low-WSF synapses as long as the classification accuracy of the network does not drop.

Testing Examples and Results

The developed pruning algorithm was tested on several classification problems (here we present two of them):

XOR Problem. It is commonly known that the XOR function (left) can be learned by structure 2-2-1 (right).

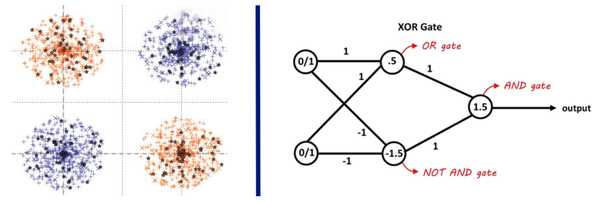


Figure: The XOR problem and its known minimal net structure.

The pruning algorithm was applied on an oversized network (2-100-1) and in 92 cases out of 100 it successfully ended up with the known minimum: 2-2-1 (6 synapses).

Recognition of Handwritten Digits. Next, we tested the method on a large MNIST dataset of 60000 samples, 10 classes and 784 features/sample. In this case, we pruned the original fully-connected net 784-20-10 (15880 synapses) and ended up with structure 465-20-10 using only **1259** active synapses, while the classification accuracy was kept on 97%. In effect, a good feature selection was performed by pruning input synapses (the following Figure - bottom left corner).

Another experiment showed that the dataset can be learned up to accuracy of 50% by a net of just 38 synapses and using 20 features only (on the right in the Figure). Here we can label parts of the net responsible for individual classes.

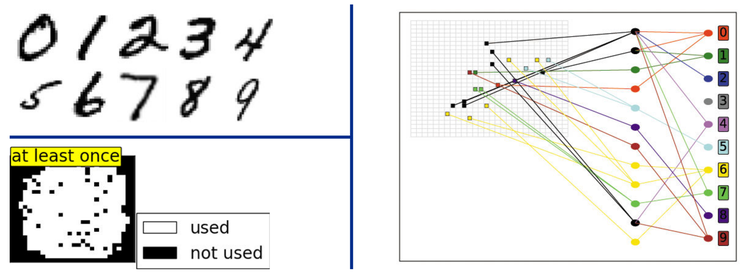


Figure: Samples of MNIST dataset (upper left), used features after pruning (bottom left), example of network demystification (right).

Summary

We developed a method, which significantly simplifies trained neural networks and have no influence on the classification performance. It could be used for understanding some of the so-far hidden behaviour. Our main outcomes are:

- in general, over **90%** of synapses are redundant;
- we end with a **minimal net structure** for given data;
- the developed method does **feature selection**;
- learning and prediction **time is rapidly reduced**;
- the method is applicable to any classification problem.

[1] Martin Bulín, MSc., author of the master thesis, email: bulinmartin@gmail.com
[2] Ing. Luboš Šmídl, Ph.D., supervisor of the master thesis, email: smidl@kky.zcu.cz

DEPARTMENT OF CYBERNETICS