Multilayer Perceptron Training Randomized by Second Instance

Todor Balabanov, Iliyan Zankinski, Kolyu Kolev

Introduction Multilayer perceptron is one of the most used types of artificial neural network. For the last four decades artificial neural networks are heavily researched and used in real industrial solutions. The power of artificial neural networks is in there operating phase. Once trained artificial neural networks are extremely efficient tool. Training phase of the artificial neural networks is the problematic part of their usage. Training is usually too slow and not always efficient enough. Multilayer perceptron is weighted directed graph organized in layers. When multilayer perceptron is fully connected each neuron from a single layer is connected with each neuron from the next layer. Neurons are the nodes in the graph where links between nodes are weighted. The calculating power of multilayer perceptron is in its weights. Finding proper velues for the weights is a global optimization problem. Each neuron collects signals as its input. Signals are coming from other neurons or from external for the multilayer perceptron environment. Signals are multiplied by the weight of the link and after that sum is calculated. Sum of the weighted signals is usually normalized by neuron's activation function. Such normalization is a key feature of the multilayer perceptron, because the number of neuron's input links is variable and there is no limitation of the weights range. The most used activation functions are the hyperbolic tangent and the sigmoid function. In the literature there are hundreds training algorithms, but the most popular and the most used one is the back-propagation training. Back-propagation is exact numerical method and it is based on the gradient of the multilayer perceptron output error. Speed-up of multilayer perceptron training is always desirable and this study proposes usage of second multilayer perceptron which randomizes the back-propagation training procedure of the basic multilayer perceptron.

Proposed Improvement

Conclusions All initial experiments are done with open source software solution, based on Encog Machine Learning Framework (https://www.heatonresearch.com/encog/). Randomization of the weights in the basic multilayer perceptron lead to stairs like convergence curve. As parallel to the neutral neural networks, such stress on the neurons is observed when the neural cells are overloaded and their performance is sensitively reduced. The proposed back-propagation training modification is very promising if it is used in hybrid implementations. As further work it will be interesting more than two different activation functions to be used. Some of the available activation functions (https://en.wikipedia.org/wiki/Activation_function) can lead to much more promising results.

Acknowledgments This work was supported by private funding of Velbazhd Software LLC.