Author's Response to the note *Proving Hardness of Neural Network Training Problems* by Michael Schmitt

Jiří Šíma

I am grateful to Michael Schmitt for his comments regarding my paper *Back-propagation is not Efficient*. Namely, he discusses and interprets the corollary of my hardness result concerning back-propagation training for networks with one layer of a given number of neurons. I agree with him that Corollary 2 does not say much more than Theorem 1 (concerning 3-node networks) and that a convincing evidence for one-hidden layer networks would require a direct elaborate reduction (similarly as it has been done by Blum and Rivest, 1992 for threshold gates). To be really accurate, this should have been explicitly mentioned in my paper and that is why I welcome M. Schmitt's detailed explanation of this issue in his note.

On the other hand, Corollary 2 is formally correct because I consider the training problem in which the number of hidden neurons is given as a part of the input. Thus the hardness result for this problem is obtained by the reduction from the NP-hard training problem for 3-node networks (Theorem 1) as the proof of Corollary 2 explicitly states and not from the *set-splitting* problem exploited in the proof of Theorem 1 as it has been probably misinterpreted by M. Schmitt (in his notation the known NP-hard problem Π is the *set-splitting* problem). In fact, this reduction is the identity (I call it perhaps ambiguously the *identical reduction*) because only the additional information supplied to the training set says that the number of hidden neurons equals two. Therefore, the above-mentioned training problem with prescribed number of hidden nodes is hard since it contains a subclass of hard instances (e.g. 3-node networks).

Similarly, other common reductions map all instances of a known hard problem only to a special subclass of possible instances of the problem being proved to be intractable. For example, proving the hardness result for the *clique* problem using a reduction from the *satisfiability* problem does not mean that every graph (clique instance) corresponds to some formula (satisfiability instance) which is mapped to it but only the graphs of a special form are used within the reduction.

Moreover, in the classical textbook of NP-completeness (Computers and Intractability by Garey and Johnson, 1979) the technique from Corollary 2, called 'the proof by restriction' here, is mentioned among three basic general NP-completeness proof types (see section 3.2 in this textbook). For example, the well-known *knapsack* problem is proved to be hard by restriction to the *partition* problem. Alternatively, for a more direct restriction consider the *hitting set* problem which is proved to be NP-complete by allowing only the instances with two-element subsets (instead of general subsets) which corresponds to the NP-complete *vertex cover* problem. The latter restriction is very similar to the one used in Corollary 2.

Formulating Corollary 2 I have aimed to express that the class of one-hidden layer networks, which is usually trained by back-propagation algorithm in practice after a fixed network topology (i.e. the number of hidden nodes) is prescribed, includes architectures (e.g. 3-node networks) which are hard to train.