For the last several centuries to solve problems in many disciplines, humans have been using the “Leonardo da Vinci” approach, which requires a detailed understanding of a problem and then designing a solution. Unfortunately as humans we are facing an increasing number of problems which we are not able to fully understand, so the design approach is becoming unfeasible. The issue is that in the computer age we are already being overwhelmed with a huge amount of data, which are very difficult for humans to understand and process. Also, there are environmental and often engineering problems, which cannot be described by equations. These very complex problems can still be solved, but the design process should be replaced by training. The approach will allow us to solve complex multidimensional nonlinear problems without the necessity of fully understanding them and the necessity of finding mathematical formulas.

The learning approach follows the following path: at first we create a very complex artificial system with random values weights, and then the system is trained for our needs. This approach relieves us, as humans, from a fully understanding the problem under investigation, but the problem can be still solved. Successful completion of the project may have a significant impact on many disciplines because it will allow us to solve many practical problems, which we have difficulties to understanding.

In order to better explain the difference between design and learning, let us consider a dog. It seems that humans will never be able to design a dog with all the neural connections in its brain. However, by replacing a complex design process by learning we can train a dog for our needs. Please notice that current progress in technology allows us to already fabricate silicon chips with 100 billion transistors which cost only tens of dollars. A neuron with sigmoidal activations function can be built out of one pair of transistors. These silicon chips are reaching such complexity that we have to start training them instead of designing.

The journey of intelligent systems started with McCulloch-Pitts Artificial Neural Networks (ANN) in 1943, which demonstrated tremendous capabilities. What was even more important, that in contrast to traditional digital design, to change the logic, there was no need to change network topology but only to adjust the values of the weights. People understood the huge capabilities of ANN, but unfortunately they did not know how to design them. Development of the Error Back Propagation EBP learning algorithm in 1989 was a fundamental breakthrough allowing training Multi-Layer Perceptron (MLP) networks. This way a design process was replaced by learning.

With the help of an EBP algorithm and its modifications, it was possible to solve many difficult problems, and the EBP breakthrough started an exponential growth in the area of intelligent systems. However, the EBP algorithm with its modifications turns out to be very slow, and hundreds of thousands of epochs were needed to solve a mid-size problem. The second breakthrough, on a slightly smaller scale, was the introduction of the second order LM (Levenberg-Marquardt) algorithm for training MLP neural networks [19, 20]. The LM algorithm increases the learning speed 100 or even 1000 times. Unfortunately, because of a necessity of computation and the storing of a larger Jacobian matrix, the LM algorithm could be used only for relatively small problems. The second limitation of the LM algorithm is that it was developed only for well-defined layer-by-layer architectures MLP, and it could not be used for more powerful learning architectures with arbitrarily connected neurons. Only very recently the new second order NBN algorithm was presented [22], in which both deficiencies of LM algorithms were solved.

In the meantime, many researchers frustrated with traditional ANN [11] moved to alternative approaches such as Support Vector Machines - SVM [5, 6] or Extreme Learning Machines – ELM [7-9]. These learning systems with shallow architectures are very fast and efficient, but in order to solve problems, an excessive number of neurons or RBF units had to be used. Resulting networks were 10 to 100 times larger than needed [27, 28]. Inefficient ways of usage of processing units (neurons) in SVM and ELM is not surprising because only weights for one output “linear neurons” are adjusted by learning algorithm parameters or weights of remaining hundreds of neurons are set by selecting “support vectors” from a training data set (SVM) or randomly selected (ELM).

Using the Parity-N problem, it was demonstrated that the capabilities of learning architectures grows linearly with network width and grows exponentially with network depth [10, 12, 35]. Therefore, with these new powerful compact architectures, it is possible to solve the same problems with 10 to 100 times smaller number of neurons than in the case of popular MLP architectures. The problem is that nobody knows how to train it efficiently. Notice that most of the development in learning algorithms was devoted to training relatively in efficient MLP systems. One of very few exceptions was the NBN algorithm [22], which is capable of training arbitrarily connected neural architectures, but even the NBN algorithm has difficulties training these architectures because of the “vanishing gradient” phenomenon [16]. There was a relatively successful attempt to train these deep MLP architectures through a combination of extensive data preprocessing, training the first layers with an unsupervised method, and training the last layers with supervised methods [30-32]. This approach, however, may not fully utilize deep architecture capabilities because, as it was mentioned before, the popular MLP systems are not that powerful. Another difficulty with this approach is a requirement of extensive involvement of humans, who are capable of merging many different techniques of artificial intelligence to solve a specific problem. Therefore, it would not be easy to spread this deep learning technology across other disciplines.

We have recently demonstrated that it is possible to use other techniques than MLP deep topology, and it is possible to efficiently train them with a dedicated gradient based second order algorithm. Our preliminary result can be analyzed by comparing Figs 5 and 10 for this proposal or to see Fig. 33 in [12], where these results were merged together. In this case, we have used our NBN algorithm developed several years ago [21, 22]. However, it should be possible to also develop a constructive algorithm for deep FCC architectures in a slightly more complicated way, which we have already done for shallow architectures [24]. Such a constructive algorithm for deep architecture could be fundamental in spreading deep learning technology across disciplines because it should be easy to use.

The proposed basic research may, therefore, change the current direction of approaching the solution of very complex problems by replacing the traditional design type of approach with learning. This way it would be possible to potentially solve very complicated problems which are too complex to be fully understood by humans. If a relatively primitive ANN with a relatively simple EBP algorithm was able to outperform the effort of many highly qualified humans then, with the proposed better architectures and better learning algorithm, we can definitely move intelligent system research on to a much higher level.