

Improving the Convergence and Extracting Useful Information from High Dimensional Big Data with Neural Networks

Waqar Ahmed Khan^{1*}, Sai-Ho Chung¹, Muhammad Usman Awan², and Xin Wen¹

¹Department of Industrial and Systems Engineering,
The Hong Kong Polytechnic University, Hung Hom, Hong Kong

²Institute of Quality and Technology Management,
University of the Punjab, Lahore, Pakistan

*Corresponding Author: waqarahmed.khan@connect.polyu.hk

Abstract

The ability of Feedforward Neural Networks (FNNs) to solve complex nonlinear high dimensional big data problems more accurately than traditional statistical techniques exist in their universal approximation capability. Recently the FNNs has gained much attention in many applications to make a more informed decision from the available information. The range of applications includes, but is not limited to, management information systems, supply chain and logistics, marketing and sales, financial analysis, product and process improvements, manufacturing cost reduction, business improvements, decision support systems, and health services. The application of the FNNs in the diverse domains are not simple and extensive knowledge is needed to achieve the desired intended results. With the advancement of modern information technology, the big data is becoming more challenging to handle because of growing and changing at a rapid rate and may become a costly resource if not processed properly. Efforts are being made to overcome the challenges by building the optimal FNN that may extract a useful pattern from the big data and generate information in real time for making better-informed decisions.

The reason which limits the applicability of FNNs in the big data is user expertise and theoretical information needed to construct the network having characteristics of better generalization performance and fast convergence. The key influencing global and local hyperparameters that need user expertise and theoretical information prior to building the network are: 1) What should be the network size and depth i.e. shallow or deep? 2) How many hidden units should be generated by each hidden layer? 3) How many hidden layers will be adequate for deep learning? 4) What should be network initial connection weights and learning rate? 5) How hyperparameters should be adjusted? 6) What should be the size of the dataset during network training? 7) Which learning algorithm should be implemented? 8) Which network typology is more efficient i.e. fixed or cascade? 9) What should be the criteria for increasing or decreasing the global and local hyperparameters? and 10) What type of activation function to be used in the hidden units?

In the literature, answers to the above questions are not straightforward. The existing applications are focused on selecting and comparing traditional algorithms which may be solely based on expertise and the available data. The insufficient user expertise and theoretical information to adjust local and global hyperparameters may cause the network to converge at suboptimal local minima with an increase in learning time. Researchers have made efforts to reduce the drawbacks by critical thinking on the above problematic questions, however, a study is missing and an open challenge to gather the answers for above questions on one platform. The purpose of this work is to answer above questions by identifying noteworthy contribution made in improving generalization performance and convergence rate of FNN, identify new research directions that will help researchers to design new, simple and efficient algorithms, and the users to implement the optimal designed FNN for solving complex nonlinear big data problems. FNN has gained much popularity during the last three decades. Therefore, the study is focused on algorithms proposed during the last three decades and their applications in solving problems in engineering, life sciences, and management sciences domains. The study identified in total 54Nos. unique learning and optimization algorithms proposed to improve the generalization performance and convergence speed of FNN. The identified learning and optimization algorithms are further classified into six categories based on their problem identification, mathematical model, technical reasoning and proposed solution. The authors also explained the importance of categories in term of their applications on more than hundred real-world problems. The categories are named as 1) Gradient learning algorithms for Network Training, 2) Optimization algorithms for learning rate, 3) Bias and variance (underfitting and overfitting) minimization algorithms, 4) Constructive topology FNNs, 5) Gradient free learning algorithms, and 6) Metaheuristic search algorithms.

The study has identified a major shift in research trend in the last three decades in improving the FNN. For instance, research contribution in FNN during the last three decades has changed from complex gradient-based algorithms to the gradient-free algorithms, trial and error hidden units fixed topology approach to the cascade topology, hyperparameters initial guess to analytically calculation, and converging algorithms at a global minimum rather than the local minimum. The identified categories may be considered as a research gap and further improvement can bring a significant contribution. The extensive knowledge in the study will contribute by helping researchers and practitioners to deeply understand FNN existing algorithms merits with limitations and research gaps. Moreover, the user, after having in-depth knowledge, can apply appropriate FNN algorithms to get optimal results at the shortest possible time with fewer efforts for their specific application and big data problems. After discussing FNN algorithms with their technical merits and limitations in their respective categories along with applications, the authors suggested five new future directions to contribute to strengthening the literature to overcome the challenges of handling big data.

Keyword: Big data, Feedforward neural networks, Generalization performance, Convergence rate