Functional Networks Trained by Optimization

Francisco Daniel Filip Duarte (✉ danielduarte1181@gmail.com)

Independent

Method Article

Keywords: neural network, response surface, functional network, model reduction

Posted Date: January 24th, 2022

DOI: <https://doi.org/10.21203/rs.3.rs-1289775/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Title: **Functional Networks Trained by Optimization**

Name: Francisco Daniel Filip Duarte

Degree: MSC in Mechanical Engineering from Concordia University, 2019.

Email: danielduarte1181@gmail.com

Website: www.filipduarte.com

Affiliation: Independent

Financial Support: No financial support provided for this study

Conflict of interest: The author declares he has a patent pending on the described functional network model.

The functional network (FN) is a patent-pending algorithm. In order to use the FN for commercial purpose please contact danielduarte1181@gmail.com.

Highlights:

- A new neural network model which allows to extract the trained information.
- Generation of a simplified mathematical expression from a pattern.
- Can be applied to simplify partitioned mathematical models which have costly computation, such as finite element models (FEM) and computer fluid dynamic (CFD) elements.

Functional Networks Trained by Optimization

F. D. Filip Duarte ¹, danielduarte1181@gmail.com

¹- **Independent**, www.filipduarte.com

Abstract: Artificial neural networks (ANNs) gained much attention due to its capacity to mimic neural cells of living organisms which have intelligence. ANNs and some of its variations, have been applied in text and image recognition, among other applications, and are the basic building blocks of artificial intelligence (AI). A known problem, however, is the impossibility to extract the knowledge achieved by training. To solve this problem, a new neural network is hereby presented, called functional network (FN). This functional network differs from the ANNs basically by the fact that the sigmoid function can be substituted by any mathematical expression, that each functional neuron has multiple inputs and multiple function outputs, and that the trained information can be extracted in the form of a mathematical expression, denominated generative mathematical expression (GME).

Although a moderate size FN may give a very large number of possible combinations, requiring very expensive computation efforts on training, recent developments in computation such as quantum computers indicate it increasingly becomes a viable alternative for large patterns and complex models. Thus, the GME of pattern representations can be extracted from a FN in order to simplify AI and numerical modeling tasks in general, such as finite element models (FEM) of structures, or computer fluid dynamic (CFD) models, decreasing overall computation costs.

Keywords: neural network, response surface, functional network, model reduction.

1. Introduction

With the development of artificial neural networks (ANNs), new research fields were created, such as text

and speech recognition. It also allowed the creation of artificial intelligence (AI), which automated several tasks which before were only accomplished by human work and cognition.

Automated AI agents perform important tasks, even with human-level intelligence or beyond, empowering the management of several complex tasks, like self-driving vehicles, stock management, warfare, and beyond. The basic building blocks of AI are ANNs, also some of its variations like convolutional neural networks (CNN), recursive neural networks (RNN), and other types of response surfaces, like Kriging (KR), support vector machine (SVM), etc. Yet some important limitations are known to these models, like the inability to extract the knowledge achieved during training.

A new type of neural network, called functional network (FN), is hereby presented. This metamodel differs from usual ANNs by the fact that any function, like a polynomial, trigonometric, or logarithmic function, for example, can substitute the activation function of the neuron cell of the model.

A second difference is that each neuron can have multiple inputs for multiple functions outputs. With this, it is possible to represent a large number of mathematical expressions in a mathematical structure similar to a neural network, facilitating the expression to be adjusted or configured. If properly adjusted and trained, the FN will provide the generative mathematical expression (GME), which represents the model or pattern.

Although a FN requires large processing capabilities for complex patterns and large GMEs, mathematical models which are often reevaluated can have their GME defined in the training of a FN, thus reducing its overall implementation cost.

2. Related work

Rumelhart et al. described the back propagation method [1], which is applied to train multi-level NNs. These ANNs consist of a mathematical model which mimics the flow of information observed in organic neuronal tissues of several living species, which allow many living organisms to have intelligence. As in the organic neuron network, a set of inputs flow through several layers of interconnected neurons. Each neuron sums all the received inputs and propagates forward the response to other neurons. The connection between each neuron has propagation weights which amplify or decrease each input, and these weights are adjusted to minimize the error of the response. These weights are adjusted by the back-propagation method, which propagates back in the network a gradient based adjustment to minimize error. This methodology has been very efficient in text, image and speech recognition, among other applications. ANNs can be applied to forecast the response of a mathematical function, and category classification.

A usual problem is that ANNs can be over trained and learn too many details over the data set used to train it, thus, losing its generalization properties. Srivastava et al. proposed a technique called dropout [2], where random neurons are cancelled during the learning process to preventing over training.

Beside ANNs there are also recurrent artificial network (RNN) [3] and convolutional neural network (CNN) [4]. RNN differs from ANN since it integrates on the forward propagation the inputs from the previous iteration. A CNN for instance, operates on image classification using different filter which allows to better process image and other datasets, which otherwise would be too complicated to process, since a usual image often has a large number of pixels.

Besides neural network type of models, other response surfaces are also applied in machine learning. Among well-known response surfaces are Kriging [5], [6], polynomial regression [7], and radial basis functions (RBF) [5], among others. In order to evaluate the efficiency of RS to forecast the output of a function, many

benchmark functions which were created [5]. Yet a common problem of these metamodels is that it is not possible to extract the knowledge information acquired during training.

Deep learning (DL) [8], is a technique created by Lecun, Bengio and Hinton, where many layers of metamodels are trained over data set to recognize several levels of patterns. It has unsupervised training, different than ANNs, and has many important applications for the automation of many complex and challenging tasks, substituting human work in many fields of activity with similar or greater levels of intelligence.

3. Methodology

The functional network has many similarities to a regular ANN. Figure 1 displays a regular ANN [1] with the input layer, the hidden layers and the output layer.

In the ANN, the input of each neuron is given by the sum of inputs of the previous layer.

Fig. 1 – The schematic of an ANN

The output of each neuron is then given by the activation function, which is applied to the input, modulating it between certain levels. This process is repeated to each neuron of each layer, which then propagates the signals to the next layer. The ANN model is trained by adjusting the weight value of each neuron connection.

Fig. 2 – The schematic of a single neuron of an ANN, where the activation function is the Sigmoid function.

At figure 2 is displayed the schematic of a single neuron, where the input X is given by the sum of the inputs multiplied by the connection weights (eq. 1).

$$X = \sum x_i k_i \quad (\text{eq. 1})$$

the output of the neuron of an ANN is given by the sigmoid function (eq. 2), or another activation function.

$$Y = F_{\text{sigmoid}}(X) \quad (\text{eq. 2})$$

The FN basically differs from an ANN by two main characteristics. First, the activation function is substituted by any function, like a polynomial, a trigonometric, logarithmic, or any other convenient mathematical expression. Secondly, each single neuron of the FN has multiples inputs and outputs of multiple functions, which operate over the same set of inputs of the same neuron. With this, a large number of mathematical expressions can be represented in a network structure.

Fig. 3 – The schematic of a single neuron of a FN

Figure 3 displays the schematic of a FN neuron, where Y_j can be an arbitrary function which operates over the set of inputs of the functional neuron (eq. 3).

$$Y_j = F_j(X_1, X_2, \dots, X_m) \quad (\text{eq. 3})$$

The FN can be trained by combinatorial optimization, in the selection of the functions of each neuron, and the adjustment of the weights of each connection, aiming to minimize the error of the response. Moderate size FNs give a very broad coordinate space, which requires significant computation efforts.

Yet with the generation of reduced mathematical expressions for patterns and models which are frequently evaluated, numerical analysis can be simplified and have overall computation costs reduced.

4. Conclusion

The FN facilitates the exploration of different networks, with different activation functions or structures. In addition, other networks with increased number of inputs and outputs can be tested, which may have greater plasticity than present models.

The main advantage of the FN, however, is that it can facilitate the search of reduced mathematical representations for many patterns, and mathematical

models which are frequently evaluated and have costly computations, such as discrete mathematical models like finite element models (FEM), finite difference model (FDM), computer fluid dynamics (CFD), and other partitioned mathematical representations. Thus, can have its formulation partially or fully simplified by a FN.

This can decrease the costs of many industrial processes of mathematical analysis, and lead numerical simulations related tasks to greater levels of performance. In addition, GME of AI patterns can be mathematically assembled or disassembled, providing greater insight and possibilities in the analysis and processing of these patterns.

5. Bibliography

- [1] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986, doi: 10.1038/323533a0.
- [2] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014.
- [3] W. Zaremba, I. Sutskever, and O. Vinyals, "Recurrent Neural Network Regularization," Sep. 2014, [Online]. Available: <http://arxiv.org/abs/1409.2329>
- [4] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *2017 International Conference on Engineering and Technology (ICET)*, 2017, pp. 1–6. doi: 10.1109/ICEngTechnol.2017.8308186.
- [5] C. Bogoclu and D. Roos, "A benchmark of contemporary metamodeling algorithms," *ECCOMAS Congress 2016 - Proceedings of the 7th European Congress on Computational Methods in Applied Sciences and Engineering*, vol. 2, no. June, pp. 3344–3360, 2016, doi: 10.7712/100016.2039.7645.
- [6] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. Cambridge, Massachusetts: MIT Press, 2006.
- [7] R. Jin, W. Chen, and T. W. Simpson, "Comparative studies of metamodeling techniques under multiple modeling criteria," *8th Symposium on Multidisciplinary Analysis and Optimization*, 2000, doi: 10.2514/6.2000-4801.
- [8] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553. Nature Publishing Group, pp. 436–444, May 2015. doi: 10.1038/nature14539.