Grey-box modeling: Training a neural network to recognize classes of inputs and identifying learned hidden layer features

Nadeem Patel *

4820 N. Central Park Ave, Chicago, IL, 60625

* nadeemp.101@gmail.com

Abstract

White-box models have observable and understandable behaviors, features, and relationships between variables and output predictions but have lower accuracy. Black-box models provide accuracy and observable input-out relationships, but there is a lack of clarity around the inner workings (Walker 2020). While both have their advantages but also have individual drawbacks. Research into grey-box modeling literature shows that a grey-box approach incorporates both white-box and black-box modeling to develop an approach that provides a balance of interpretability and accuracy. Such gray-box models systematically use partial a priori knowledge of the process and data from an experiment (Sohlberg 1998, 7)

This particular study analyzes a neural network that has a grey-box subnet, where the grey-box is pre-trained to conduct blockier, chunkier, and simpler classification. The output of this grey-box subnet becomes part of the inputs into a more complex neural network. Data used in this study includes an 81-element input grids, and these grids are comprised in a pattern to represent a letter of the alphabet. The letters are classified into "big shape groups." For example, the letters "O" and "Q" could be classified into the same class. The goal is to train a network to recognize classes of inputs by studying how hidden nodes learned to recognize specific features in similar inputs. Based on previous experimentations, separate neural networks have different connection weights unless they are identical in structure and have identical seeds for starting weights, as well as identical order for the same data presentation.

Introduction

Machine learning has seen success over the years in numerous areas when it comes to pattern recognition, including text and image recognition, object classification, and personal identification. However, the applications of machine learning often rely on black-box modeling, a non-parametric approach. On the other hand, mathematical modeling, simulation, and optimization is dependent on white-box methods for parametric models. Grey-box models combine qualitative prior knowledge with quantitative data, and this approach utilizes existing information regarding a process to determine the best possible model [see Figure 1 in Appendix] (Bortz n.d.). Ultimately, a black-box model approach provides accuracy and observable input-out relationships, but there is a lack of clarity around the inner workings. White-box models have observable and understandable behaviors, features, and relationships between variables and output predictions, but there is lower accuracy (Walker 2020). Through a grey-box approach, there is partial understanding of the internal working structure while quality results are obtained.

This study is conducted to illustrate the use of a grey-box approach in a complex neural network. The goal is to not just identify learning features in a hidden layer, but also to demonstrate how hidden nodes learn to recognize specific features. Through this approach, functional steps can be identified, and this in-depth understanding can give the ability to modify data if necessary while still obtaining high accuracy common in black-box modeling. Identifying learned hidden layer features can help to improve results. However, there are questions that need to be answered, including how hidden layer neurons deal with and represent inputs. Another essential part of the research is to explain what elements in a data set mean and how they are associated with outputs, as well as how the output classes can be configured.

Data scientists and business leaders using machine learning models and artificial intelligence systems constantly face the challenge of balancing interpretability and accuracy that comes from the differences of black-box and white-box models (Walker 2020). A grey-box model can help to find that balance through the development of an interpretable and accurate machine learning model. More importantly, it is crucial to understand how a grey-box model that is trained to recognize classes or features can be used to help a neural network make fine-tuned decisions.

Literature Review

Considering grey-box models combine the advantages of white-box and black-box components, grey-box models have been highlighted in an extensive number of research studies. Through the study of grey-box models, the disadvantages of black-box and white-box methods are pointed out as well. A process model based on complete knowledge of a process is conventionally viewed as a white-box model while a model based on experimental data is referred to as a black-box model. When comparing such models to grey-box modeling, the difference is the systematic use of partial a priori knowledge of the process and data from an experiment (Sohlberg 1998, 7).

Grey-box modeling has been utilized for chemical engineering processes, pattern recognition in process data, and forming technology (Bortz n.d.). This idea is further emphasized in another study that states grey-box models have been mainly used to construct biochemical and chemical systems, which are usually spatially distributed in their nature due to physical processes such as heat transfer, diffusion, or advection. Grey-box models are applied to such systems under the assumption that the processes can be ignored, and a grey-box method is used to model distributed parameter systems, which are systems whose state space cannot be fully described by a finite number of parameters. According to the study, the grey-box model was able to successfully capture the dynamics of the reaction system for one-dimensional and twodimensional systems. While the method was sensitive to noise, the method is considered viable for modelling the reaction system in the study (Barkman 2018, 54).

When looking at the high-level research studies, it becomes clear how accuracy and interpretability are both achieved. By highlighting the benefits and flaws of white-box and black-box models, the studies can develop a process of finding the ideal approach to incorporate both methods. For example, a study looked to develop and validate a grey-box model for refrigeration applications. The paper discussed the main characteristics of white-box, black-box, and their integration into grey-box models, as well as the requirement and sourcing of accurate data for model development and essential validation concepts. According to the study, white-box and black-box models represent two different modeling philosophies, and the two philosophies are combined in grey-box models, which help to preserve the benefits of the fundamental approach of white-box models with the data-based approach of black-box models measures [see Figure 2 in Appendix] (Estrada-Flores et al. 2006, 932). Similarly, another study aimed to develop an interpretable and accurate machine learning model. The grey-box model was based on a semi-supervised methodology utilizing a self-training framework, and the results of the model were compared to those of a black-box and white-box models. The proposed grey-box model had the accuracy comparable to that of a black-box model and better accuracy compared to a white-box model (Pintelas et al. 2020).

Several literature pieces focus on the non-linear parts of a grey-box model as it pertains to achieving a better model of the process. In a grey-box model, it is possible to build in non-linearities and separate different parts of the process into submodels. When considering the possibility of being able to divide a process model into submodels or internal models, the model achieved through a grey-box approach can be used to optimize the performance of the system (Sohlberg 1998, 9).

Penetration testing utilizes black-box, white-box, and grey-box approaches, and this is another major area of focus in literature. Much like other implementations, white-box test has full knowledge of client network and applications, including code, network diagrams, and system access. This approach provides a comprehensive approach, so there are no limits to the bugs that can be found. Also, such a test can help to prevent bugs before something is pushed to production. However, like other applications, one can easily overlook certain bugs through white-box testing. With black-box testing, there is no knowledge of client network and applications, but it emulates real-world threats. While this type of modeling provides a faster and simplistic approach, it is not as comprehensive as white-box modeling, so some bugs get missed. Finally, a grey-box test is a combination of white-box and black-box testing and provides some knowledge of internal systems. While this approach provides a holistic coverage when it comes to testing, it does not master any specific type of testing. It is slower than black-box testing and not as comprehensive as white-box testing (Lowrie 2021).

Like other state-of-the-art research, this study also looks to highlight the advantages of grey-box models by training a neural network to recognize ("discriminate amongst") inputs by classifying them into desired output classes. More importantly, this study assesses hidden node activations after the network is trained. Other studies took similar approaches by not just implementing a grey-box model, but also experimenting with parameters and analyzed the outputs. Similar to this research, a study used a grey-box setting and targeted models with a pre-trained encoder followed by a single classification layer. According to that study, when an encoder is fine-tuned, a hybrid learning-based and algebraic attack improves over the learning-based state-of-the-art without requiring additional queries (Zanella-Bengelin et al. 2021). Several studies also used white or random noise for training data, an approach also utilized in this study.

Data

The data used in this research is not selected from external sources, so there was no need for preprocessing or other necessary steps to organize collected information. Instead, the data was curated of 16 sets for the first 16 instances of the alphabets, and each data set has six elements. Since the code is written in Python, first element is actually viewed as the zeroth element in the number of the training data set, and this element starts at one and goes up to the total number of training instances used. This numeric value corresponds to the letter in the alphabet. For example, A is one since A is the first letter. The first element is a nine-by-nine input “grid,” so this can be viewed as an 81-element list made up of zeroes and ones within a square bracket for the specified letter. The grid pattern is formed to represent the specified letter. The second element is the number corresponding to the letter of the alphabet for the output class, and there can only be 26 different output classes. Like the zeroth element, this element also starts at one and goes up to the number of training instances used. The third element is the actual letter of the alphabet corresponding to the particular training data set, and this represents both primary and variants on a single letter. The fourth element is the number corresponding to the “big shape class” that the letter would be a part of. There are typically only half as many – or less – “big shape classes” than there are letters. The classes start numbering at one, which is not Python-style. The final element is the letter corresponding to the “big shape class” that the letter will be a part of. For example, “Q” would fall into the “O” class or “H” would fall into “A” class. These classes can be changed or additional classes can be added. [See Figure 4 in Appendix.]

The data is used to determine how closely error values align with desired outputs. Since each input letter has a desired output, which is a “big shape class,” the data will output results that provide actual outputs and error values. By understanding the true results based on how the parameters are configured, insights can be drawn on how and which parameters to fine tune in order to obtain the best results.

Methods

The method to conduct research consists of searching for reliable literature focused on grey-box models. Through the proper understanding of grey-box models, they can be applied to a neural network appropriately. In this research, the initial set-up will include a neural network that looks to recognize broader classes of letters like big circular letters or large-scale features. That first step of training occurs within the grey box that is inserted in the neural network as a pre-trained subnet. The grey box has stored weights as a component to help the neural network make fine-tuned decisions about the bigger classification problem. The outputs of the subnet, a grey-box network, become the inputs to the hidden layer of the larger network [see Figure 3 in Appendix].

The hidden node outputs are to be assessed to determine what features the nodes respond to and correlate the features with similar characteristics of letters in the data set. Based on the hidden node responses and similar features, the big shape classes are identified, and the previous classification of a given letter per class are replaced with a smaller number of classes that have common “big shapes.”

Regarding the code, the parameters that will be manipulated are Alpha, the steepness of the transfer function, Eta, the learning rate, maxNumIterations, the total number of iterations, and

Epsilon, the criteria for accepting that the network has been sufficiently trained. Manipulating the parameters to obtain desired results can help to gain a deeper understanding of the inner workings of the gray-box model.

Results

Table 1

Alpha = 1.0 Eta = 0.5 maxNumIterations = 5000 Epsilon = 0.01

Input Letter (Third Element)	Desired Output (Ffith Element)	Desired Output Corresponding Number (Fourth Element)	Hidden Node Activations	Output Node Activations	Desired Output Array Values	Error Values	New SSE
A	A	0	[0.72 0.01 0.41 0.55 0.02 0.33]	[**0.60** 0.73 0.19 0.28 0.45 0.36 0.52 0.57 0.47]	[1. 0. 0. 0. 0. 0. 0. 0. 0.]	[**0.30** -0.73 -0.19 -0.28 -0.45 -0.36 -0.52 -0.57 -0.45]	1.94
B	B	1	[0.76 0.00 0.08 0.33 0.00 0.96]	[0.62 **0.91** 0.13 0.51 0.71 0.41 0.49 0.62 0.27]	[0. 1. 0. 0. 0. 0. 0. 0. 0.]	[-0.62 **0.09** -0.13 -0.51 -0.71 -0.41 -0.49 -0.62 -0.27]	2.03
C	C	2	[0.05 0.86 0.03 0.58 0.95 0.87]	[0.23 0.64 **0.71** 0.47 0.58 0.78 0.66 0.64 0.76]	[0. 0. 1. 0. 0. 0. 0. 0. 0.]	[-0.23 -0.64 **0.29** -0.47 -0.58 -0.78 -0.66 -0.64 -0.76]	3.13
D	O	3	[0.69 0.91 0.03 0.92 0.98 0.27]	[0.14 0.33 0.54 **0.36** 0.40 0.78 0.45 0.53 0.77]	[0. 0. 0. 1. 0. 0. 0. 0. 0.]	[-0.14 -0.33 -0.54 **0.65** -0.40 -0.78 -0.45 -0.53 -0.77]	2.67
E	E	4	[0.05 0.86 0.01 0.05 0.95 0.92]	[0.32 0.75 0.80 0.55 **0.64** 0.73 0.73 0.73 0.78]	[0. 0. 0. 0. 1. 0. 0. 0. 0.]	[-0.32 -0.75 -0.80 -0.55 **0.36** -0.73 -0.73 -0.73 -0.78]	3.94
I	I	5	[0.71 0.27 0.03 0.72 0.38 0.46]	[0.35 0.30 0.53 0.36 0.51 **0.73** 0.44 0.38 0.82]	[0. 0. 0. 0. 0. 1. 0. 0. 0.]	[-0.35 -0.30 -0.53 -0.36 -0.51 **0.27** -0.44 -0.38 -0.82]	1.95
K	K	6	[0.93 0.42 0.20 0.95 0.63 0.10]	[0.37 0.88 0.09 0.58 0.71 0.21 **0.43** 0.87 0.27]	[0. 0. 0. 0. 0. 0. 1. 0. 0.]	[-0.37 -0.88 -0.09 -0.58 -0.71 -0.21 **0.58** -0.86 -0.27]	2.94
L	L	7	[0.93 0.01 0.02 0.91 0.09 0.30]	[0.30 0.44 0.57 0.28 0.68 0.06 0.83 **0.70** 0.28]	[0. 0. 0. 0. 0. 0. 0. 1. 0.]	[-0.38 -0.87 -0.09 -0.51 -0.66 -0.17 -0.45 **0.16** -0.28]	1.94
M	M	8	[0.03 0.00 0.99 0.99 0.98 0.00]	[0.05 0.00 0.08 0.03 0.00 0.00 0.00 0.04 **0.90**]	[0. 0. 0. 0. 0. 0. 0. 0. 1.]	[-0.05 -0.00 -0.08 -0.03 -0.00 -0.00 -0.00 -0.04 **0.10**]	0.02

Table 2

Alpha = 1.0 Eta = 0.5 maxNumIterations = 5000 Epsilon = 0.1

Input Letter (Third Element)	Desired Output (Ffith Element)	Desired Output Corresponding Number (Fourth Element)	Hidden Node Activations	Output Node Activations	Desired Output Array Values	Error Values	New SSE
A	A	0	[0.12 1.00 0.65 0.06 0.49 0.04]	[**0.41** 0.37 0.11 0.51 0.51 0.59 0.83 0.71 0.90]	[1. 0. 0. 0. 0. 0. 0. 0. 0.]	[**0.59** -0.37 -0.11 -0.51 -0.51 -0.59 -0.83 -0.71 -0.90]	3.36
B	B	1	[0.98 0.16 0.91 0.66 0.01 0.26]	[0.49 **0.40** 0.31 0.30 0.19 0.70 0.66 0.66 0.87]	[0. 1. 0. 0. 0. 0. 0. 0. 0.]	[-0.49 **0.60** -0.31 -0.30 -0.19 -0.70 -0.66 -0.66 -0.87]	2.92
C	C	2	[0.83 0.06 0.64 0.29 0.00 0.46]	[0.56 0.31 **0.24** 0.29 0.20 0.67 0.53 0.69 0.80]	[0. 0. 1. 0. 0. 0. 0. 0. 0.]	[-0.56 -0.31 **0.76** -0.29 -0.20 -0.67 -0.53 -0.69 -0.80]	2.96
D	O	3	[0.87 0.04 0.76 0.44 0.00 0.18]	[0.51 0.40 0.31 **0.30** 0.21 0.68 0.60 0.66 0.84]	[0. 0. 0. 1. 0. 0. 0. 0. 0.]	[-0.51 -0.40 -0.31 **0.70** -0.21 -0.68 -0.60 -0.66 -0.84]	3.01
E	E	4	[0.93 0.58 0.65 0.56 0.01 0.59]	[0.48 0.28 0.21 0.37 **0.24** 0.71 0.68 0.65 0.83]	[0. 0. 0. 0. 1. 0. 0. 0. 0.]	[-0.48 -0.28 -0.21 -0.37 **0.76** -0.71 -0.68 -0.65 -0.83]	3.13
I	I	5	[0.62 0.36 0.25 0.95 0.03 0.78]	[0.36 0.30 0.26 0.34 0.20 **0.56** 0.70 0.62 0.74]	[0. 0. 0. 0. 0. 1. 0. 0. 0.]	[-0.36 -0.30 -0.26 -0.34 -0.20 **0.45** -0.70 -0.62 -0.74]	2.05
K	K	6	[0.04 0.40 0.32 0.18 0.16 0.05]	[0.47 0.18 0.16 0.40 0.25 0.65 **0.63** 0.65 0.73]	[0. 0. 0. 0. 0. 0. 1. 0. 0.]	[-0.39 -0.40 -0.20 -0.38 -0.36 -0.53 **0.31** -0.64 -0.80]	2.07
L	L	7	[0.79 0.02 0.90 0.24 0.01 0.59]	[0.62 0.30 0.20 0.22 0.17 0.70 0.48 **0.73** 0.82]	[0. 0. 0. 0. 0. 0. 0. 1. 0.]	[-0.62 -0.30 -0.19 -0.22 -0.17 -0.70 -0.48 **0.27** -0.82]	2.06
M	M	8	[0.03 0.14 0.70 0.01 0.01 0.01]	[0.50 0.43 0.20 0.24 0.30 0.61 0.56 0.67 **0.82**]	[0. 0. 0. 0. 0. 0. 0. 0. 1.]	[-0.50 -0.43 -0.20 -0.24 -0.27 -0.61 -0.56 -0.67 **0.18**]	1.78

Table 3

Alpha = 1.0 Eta = 0.5 maxNumIterations = 5000 Epsilon = 0.2

Input Letter (Third Element)	Desired Output (Ffith Element)	Desired Output Corresponding Number (Fourth Element)	Hidden Node Activations	Output Node Activations	Desired Output Array Values	Error Values	New SSE
A	A	0	[0.24 0.95 0.30 0.76 0.95 0.35]	[**0.82** 0.55 0.16 0.59 0.47 0.60 0.58 0.32 0.46]	[1. 0. 0. 0. 0. 0. 0. 0. 0.]	[**0.18** -0.55 -0.16 -0.59 -0.47 -0.60 -0.58 -0.32 -0.46]	1.95
B	B	1	[0.24 0.54 0.75 0.04 0.10 0.27]	[0.79 **0.53** 0.12 0.59 0.57 0.71 0.53 0.49 0.50]	[0. 1. 0. 0. 0. 0. 0. 0. 0.]	[-0.79 **0.47** -0.12 -0.59 -0.57 -0.71 -0.53 -0.49 -0.50]	2.82
C	C	2	[0.76 0.93 0.64 0.17 0.79 0.10]	[0.76 0.59 **0.19** 0.72 0.52 0.51 0.55 0.30 0.49]	[0. 0. 1. 0. 0. 0. 0. 0. 0.]	[-0.76 -0.59 **0.81** -0.72 -0.52 -0.51 -0.55 -0.30 -0.49]	3.25
D	O	3	[0.86 0.75 0.31 0.02 0.97 0.03]	[0.69 0.58 0.18 **0.75** 0.51 0.54 0.64 0.29 0.55]	[0. 0. 0. 1. 0. 0. 0. 0. 0.]	[-0.69 -0.58 -0.18 **0.25** -0.51 -0.54 -0.64 -0.29 -0.55]	2.25
E	E	4	[0.43 0.92 0.66 0.20 0.84 0.55]	[0.84 0.53 0.14 0.71 **0.51** 0.59 0.58 0.40 0.55]	[0. 0. 0. 0. 1. 0. 0. 0. 0.]	[-0.84 -0.53 -0.14 -0.71 **0.49** -0.59 -0.58 -0.40 -0.55]	2.90
I	I	5	[0.95 0.90 0.79 0.12 0.26 0.39]	[0.79 0.57 0.28 0.76 0.47 **0.39** 0.52 0.35 0.52]	[0. 0. 0. 0. 0. 1. 0. 0. 0.]	[-0.79 -0.57 -0.28 -0.76 -0.47 **0.61** -0.52 -0.35 -0.52]	2.87
K	K	6	[0.77 0.65 0.09 0.10 0.92 0.40]	[0.76 0.64 0.24 0.67 0.34 0.55 **0.62** 0.24 0.45]	[0. 0. 0. 0. 0. 0. 1. 0. 0.]	[-0.76 -0.64 -0.24 -0.67 -0.34 -0.55 **0.38** -0.24 -0.45]	2.32
L	L	7	[0.65 0.77 0.29 0.83 0.99 0.24]	[0.77 0.63 0.21 0.64 0.41 0.57 0.58 **0.25** 0.43]	[0. 0. 0. 0. 0. 0. 0. 1. 0.]	[-0.77 -0.63 -0.21 -0.64 -0.41 -0.57 -0.58 **0.75** -0.43]	3.02
M	M	8	[0.75 0.20 0.00 0.41 1.00 0.21]	[0.66 0.58 0.21 0.66 0.42 0.64 0.66 0.35 **0.53**]	[0. 0. 0. 0. 0. 0. 0. 0. 1.]	[-0.66 -0.58 -0.21 -0.66 -0.42 -0.64 -0.66 -0.35 **0.47**]	2.62

Table 4

Alpha = 1.0 Eta = 0.5 maxNumIterations = 5000 Epsilon = 0.3

Input Letter (Third Element)	Desired Output (Ffith Element)	Desired Output Corresponding Number (Fourth Element)	Hidden Node Activations	Output Node Activations	Desired Output Array Values	Error Values	New SSE
A	A	0	[0.39 0.02 0.14 0.46 0.81 0.48]	[**0.59** 0.43 0.54 0.34 0.49 0.80 0.44 0.54 0.22]	[1. 0. 0. 0. 0. 0. 0. 0. 0.]	[**0.41** -0.43 -0.54 -0.34 -0.49 -0.80 -0.44 -0.54 -0.22]	2.15
B	B	1	[0.00 0.87 0.19 0.00 1.00 0.11]	[0.71 **0.34** 0.44 0.49 0.42 0.82 0.67 0.54 0.22]	[0. 1. 0. 0. 0. 0. 0. 0. 0.]	[-0.71 **0.66** -0.44 -0.49 -0.42 -0.82 -0.67 -0.54 -0.22]	3.00
C	C	2	[0.00 0.99 0.12 0.00 0.99 0.03]	[0.70 0.29 **0.43** 0.49 0.43 0.82 0.70 0.52 0.22]	[0. 0. 1. 0. 0. 0. 0. 0. 0.]	[-0.70 -0.29 **0.57** -0.49 -0.43 -0.82 -0.70 -0.52 -0.22]	2.81
D	O	3	[0.00 0.73 0.04 0.01 0.99 0.10]	[0.71 0.34 0.48 **0.45** 0.45 0.82 0.68 0.51 0.25]	[0. 0. 0. 1. 0. 0. 0. 0. 0.]	[-0.71 -0.33 -0.48 **0.55** -0.45 -0.82 -0.68 -0.51 -0.25]	2.81
E	E	4	[0.00 0.70 0.03 0.00 1.00 0.01]	[0.69 0.33 0.49 0.44 **0.43** 0.83 0.68 0.50 0.26]	[0. 0. 0. 0. 1. 0. 0. 0. 0.]	[-0.69 -0.33 -0.49 -0.44 **0.57** -0.83 -0.68 -0.50 -0.26]	2.81
I	I	5	[0.04 0.92 0.40 0.21 1.00 0.74]	[0.76 0.45 0.36 0.50 0.48 **0.78** 0.64 0.63 0.13]	[0. 0. 0. 0. 0. 1. 0. 0. 0.]	[-0.76 -0.45 -0.36 -0.50 -0.48 **0.22** -0.64 -0.63 -0.13]	2.26
K	K	6	[0.12 0.97 0.94 0.01 0.81 0.90]	[0.74 0.60 0.35 0.60 0.36 0.73 **0.53** 0.74 0.11]	[0. 0. 0. 0. 0. 0. 1. 0. 0.]	[-0.74 -0.56 -0.35 -0.60 -0.36 -0.73 **0.47** -0.74 -0.11]	2.78
L	L	7	[0.01 0.92 0.28 0.01 0.96 0.39]	[0.74 0.39 0.42 0.50 0.44 0.79 0.67 **0.59** 0.19]	[0. 0. 0. 0. 0. 0. 0. 1. 0.]	[-0.74 -0.39 -0.42 -0.50 -0.44 -0.79 -0.67 **0.41** -0.19]	2.6
M	M	8	[0.63 0.18 0.43 0.00 0.10 0.26]	[0.41 0.30 0.56 0.48 0.41 0.62 0.35 0.64 **0.35**]	[0. 0. 0. 0. 0. 0. 0. 0. 1.]	[-0.41 -0.30 -0.56 -0.48 -0.41 -0.62 -0.35 -0.64 **0.65**]	2.31

Table 5

Alpha = 0.5 Eta = 0.5 maxNumIterations = 5000 Epsilon = 0.3

Input Letter (Third Element)	Desired Output (Ffith Element)	Desired Output Corresponding Number (Fourth Element)	Hidden Node Activations	Output Node Activations	Desired Output Array Values	Error Values	New SSE
A	A	0	[0.43 0.71 0.12 0.33 0.80 0.86]	[**0.42** 0.54 0.50 0.48 0.63 0.48 0.44 0.63 0.64]	[1. 0. 0. 0. 0. 0. 0. 0. 0.]	[**0.58** -0.54 -0.50 -0.48 -0.63 -0.48 -0.44 -0.63 -0.64]	2.74
B	B	1	[0.25 0.21 0.30 0.42 0.76 0.96]	[0.43 **0.52** 0.57 0.53 0.63 0.44 0.44 0.61 0.70]	[0. 1. 0. 0. 0. 0. 0. 0. 0.]	[-0.43 **0.48** -0.57 -0.53 -0.63 -0.44 -0.44 -0.61 -0.70]	2.66
C	C	2	[0.04 0.40 0.22 0.33 0.77 0.95]	[0.40 0.52 **0.54** 0.51 0.60 0.48 0.46 0.62 0.68]	[0. 0. 1. 0. 0. 0. 0. 0. 0.]	[-0.40 -0.52 **0.46** -0.51 -0.60 -0.48 -0.46 -0.62 -0.68]	2.54
D	O	3	[0.11 0.41 0.15 0.15 0.88 0.97]	[0.41 0.51 0.55 **0.48** 0.59 0.49 0.43 0.62 0.67]	[0. 0. 0. 1. 0. 0. 0. 0. 0.]	[-0.41 -0.51 -0.55 **0.52** -0.59 -0.49 -0.43 -0.62 -0.67]	2.62
E	E	4	[0.15 0.48 0.26 0.29 0.75 0.96]	[0.41 0.52 0.53 0.49 **0.61** 0.47 0.46 0.63 0.68]	[0. 0. 0. 0. 1. 0. 0. 0. 0.]	[-0.41 -0.52 -0.53 -0.49 **0.39** -0.47 -0.46 -0.63 -0.68]	2.41
I	I	5	[0.24 0.19 0.15 0.29 0.37 0.78]	[0.48 0.56 0.55 0.53 0.57 **0.46** 0.46 0.61 0.68]	[0. 0. 0. 0. 0. 1. 0. 0. 0.]	[-0.43 -0.56 -0.55 -0.53 -0.57 **0.54** -0.46 -0.61 -0.68]	2.75
K	K	6	[0.10 0.19 0.58 0.40 0.81 0.75]	[0.45 0.46 0.56 0.57 0.62 0.46 **0.45** 0.63 0.69]	[0. 0. 0. 0. 0. 0. 1. 0. 0.]	[-0.45 -0.46 -0.56 -0.57 -0.62 -0.46 **0.55** -0.63 -0.69]	2.84
L	L	7	[0.11 0.56 0.19 0.32 0.88 0.89]	[0.40 0.51 0.52 0.50 0.61 0.49 0.45 **0.63** 0.66]	[0. 0. 0. 0. 0. 0. 0. 1. 0.]	[-0.40 -0.51 -0.52 -0.50 -0.61 -0.49 -0.45 **0.37** -0.66]	2.32
M	M	8	[0.58 0.68 0.33 0.11 0.84 0.95]	[0.45 0.52 0.52 0.45 0.65 0.47 0.43 0.64 **0.65**]	[0. 0. 0. 0. 0. 0. 0. 0. 1.]	[-0.45 -0.52 -0.52 -0.45 -0.65 -0.47 -0.43 -0.64 **0.35**]	2.30

Table 6

Alpha = 1.5 Eta = 0.5 maxNumIterations = 5000 Epsilon = 0.3

Input Letter (Third Element)	Desired Output (Ffith Element)	Desired Output Corresponding Number (Fourth Element)	Hidden Node Activations	Output Node Activations	Desired Output Array Values	Error Values	New SSE
A	A	0	[0.47 0.03 0.00 0.16 0.51 0.30]	[**0.25** 0.71 0.11 0.65 0.38 0.91 0.60 0.55 0.41]	[1. 0. 0. 0. 0. 0. 0. 0. 0.]	[**0.75** -0.71 -0.11 -0.65 -0.38 -0.91 -0.60 -0.55 -0.41]	3.30
B	B	1	[1.00 0.00 0.00 0.01 0.50 0.62]	[0.28 **0.61** 0.08 0.58 0.28 0.95 0.63 0.41 0.34]	[0. 1. 0. 0. 0. 0. 0. 0. 0.]	[-0.28 **0.39** -0.08 -0.58 -0.28 -0.95 -0.63 -0.41 -0.34]	2.23
C	C	2	[0.97 0.01 0.01 0.00 0.01 0.79]	[0.31 0.38 **0.07** 0.57 0.22 0.91 0.59 0.31 0.27]	[0. 0. 1. 0. 0. 0. 0. 0. 0.]	[-0.31 -0.38 **0.93** -0.57 -0.22 -0.91 -0.59 -0.31 -0.27]	2.82
D	O	3	[0.94 0.03 0.00 0.00 0.04 0.85]	[0.28 0.37 0.07 **0.59** 0.22 0.90 0.62 0.31 0.30]	[0. 0. 0. 1. 0. 0. 0. 0. 0.]	[-0.28 -0.37 -0.07 **0.41** -0.22 -0.90 -0.62 -0.31 -0.30]	1.82
E	E	4	[0.94 0.03 0.20 0.05 0.01 0.83]	[0.32 0.42 0.05 0.63 **0.19** 0.90 0.62 0.33 0.29]	[0. 0. 0. 0. 1. 0. 0. 0. 0.]	[-0.32 -0.42 -0.05 -0.63 **0.81** -0.90 -0.62 -0.33 -0.29]	2.72
I	I	5	[0.53 0.07 0.02 0.69 0.68 0.73]	[0.15 0.59 0.06 0.78 0.29 **0.91** 0.65 0.40 0.72]	[0. 0. 0. 0. 0. 1. 0. 0. 0.]	[-0.15 -0.59 -0.06 -0.78 -0.29 **0.09** -0.65 -0.40 -0.72]	2.17
K	K	6	[0.01 0.03 0.00 0.14 0.10 0.67]	[0.17 0.48 0.17 0.73 0.37 0.74 **0.68** 0.53 0.50]	[0. 0. 0. 0. 0. 0. 1. 0. 0.]	[-0.17 -0.48 -0.17 -0.73 -0.37 -0.74 **0.32** -0.53 -0.50]	2.13
L	L	7	[0.90 0.18 0.01 0.01 0.75 0.38]	[0.24 0.73 0.10 0.61 0.31 0.95 0.69 **0.49** 0.39]	[0. 0. 0. 0. 0. 0. 0. 1. 0.]	[-0.24 -0.73 -0.10 -0.61 -0.31 -0.95 -0.69 **0.51** -0.39]	2.87
M	M	8	[0.99 0.00 0.02 0.00 0.30 0.40]	[0.38 0.63 0.07 0.52 0.28 0.95 0.53 0.43 **0.23**]	[0. 0. 0. 0. 0. 0. 0. 0. 1.]	[-0.38 -0.63 -0.07 -0.52 -0.28 -0.95 -0.53 -0.43 **0.77**]	2.85

Table 7

Alpha = 2.0 Eta = 0.5 maxNumIterations = 5000 Epsilon = 0.3

Input Letter (Third Element)	Desired Output (Ffith Element)	Desired Output Corresponding Number (Fourth Element)	Hidden Node Activations	Output Node Activations	Desired Output Array Values	Error Values	New SSE
A	A	0	[0.93 0.01 0.00 0.02 0.91 0.40]	[**0.53** 0.63 0.79 0.55 0.30 0.98 0.44 0.89 0.23]	[1. 0. 0. 0. 0. 0. 0. 0. 0.]	[**0.47** -0.63 -0.79 -0.55 -0.30 -0.98 -0.44 -0.89 -0.23]	3.64
B	B	1	[0.18 0.56 0.00 0.00 0.83 0.64]	[0.47 **0.53** 0.93 0.09 0.52 0.94 0.42 0.85 0.20]	[0. 1. 0. 0. 0. 0. 0. 0. 0.]	[-0.47 **0.47** -0.93 -0.09 -0.52 -0.94 -0.42 -0.85 -0.20]	3.41
C	C	2	[0.00 0.79 0.00 0.00 0.03 0.95]	[0.13 0.65 **0.96** 0.16 0.83 0.69 0.24 0.94 0.05]	[0. 0. 1. 0. 0. 0. 0. 0. 0.]	[-0.13 -0.65 **0.04** -0.16 -0.83 -0.69 -0.24 -0.94 -0.05]	2.57
D	O	3	[0.78 0.08 0.00 0.13 0.02 0.06]	[0.43 0.57 0.61 **0.82** 0.50 0.91 0.13 0.96 0.06]	[0. 0. 0. 1. 0. 0. 0. 0. 0.]	[-0.43 -0.57 -0.61 **0.18** -0.50 -0.91 -0.13 -0.96 -0.06]	2.93
E	E	4	[0.01 0.88 0.00 0.00 0.02 1.00]	[0.10 0.65 0.97 0.14 **0.82** 0.67 0.24 0.94 0.04]	[0. 0. 0. 0. 1. 0. 0. 0. 0.]	[-0.10 -0.65 -0.97 -0.14 **0.18** -0.67 -0.24 -0.94 -0.04]	2.82
I	I	5	[1.00 0.95 0.00 0.07 0.61 0.28]	[0.14 0.42 0.88 0.29 0.11 **0.94** 0.22 0.97 0.04]	[0. 0. 0. 0. 0. 1. 0. 0. 0.]	[-0.14 -0.42 -0.88 -0.29 -0.11 **0.06** -0.22 -0.97 -0.04]	2.05
K	K	6	[0.51 0.00 0.00 0.01 0.87 0.11]	[0.80 0.48 0.70 0.41 0.40 0.98 **0.37** 0.83 0.33]	[0. 0. 0. 0. 0. 0. 1. 0. 0.]	[-0.80 -0.48 -0.70 -0.41 -0.40 -0.98 **0.63** -0.83 -0.33]	3.85
L	L	7	[0.00 0.08 0.00 0.01 0.98 0.19]	[0.89 0.43 0.77 0.15 0.58 0.97 0.41 **0.72** 0.47]	[0. 0. 0. 0. 0. 0. 0. 1. 0.]	[-0.89 -0.43 -0.77 -0.15 -0.58 -0.97 -0.41 **0.28** -0.47]	3.34
M	M	8	[0.89 0.08 0.00 0.05 1.00 0.00]	[0.72 0.46 0.65 0.49 0.18 0.99 0.36 0.88 **0.25**]	[0. 0. 0. 0. 0. 0. 0. 0. 1.]	[-0.72 -0.46 -0.65 -0.49 -0.18 -0.99 -0.36 -0.88 **0.75**]	3.86

Table 8

Alpha = 1.5 Eta = 0.5 maxNumIterations = 1000 Epsilon = 0.3

Input Letter (Third Element)	Desired Output (Ffith Element)	Desired Output Corresponding Number (Fourth Element)	Hidden Node Activations	Output Node Activations	Desired Output Array Values	Error Values	New SSE
A	A	0	[0.99 1.00 0.16 0.99 0.12 0.97]	[**0.18** 0.13 0.22 0.06 0.03 0.61 0.23 0.31 0.15]	[1. 0. 0. 0. 0. 0. 0. 0. 0.]	[**0.83** -0.13 -0.22 -0.06 -0.03 -0.61 -0.23 -0.31 -0.15]	1.29
B	B	1	[1.00 1.00 1.00 1.00 0.18 0.57]	[0.39 **0.25** 0.14 0.02 0.02 0.26 0.30 0.14 0.09]	[0. 1. 0. 0. 0. 0. 0. 0. 0.]	[-0.39 **0.75** -0.14 -0.02 -0.02 -0.26 -0.30 -0.14 -0.09]	0.92
C	C	2	[0.97 0.98 0.96 1.00 0.70 0.88]	[0.33 0.24 **0.10** 0.02 0.02 0.34 0.29 0.11 0.15]	[0. 0. 1. 0. 0. 0. 0. 0. 0.]	[-0.33 -0.24 **0.90** -0.02 -0.02 -0.34 -0.29 -0.11 -0.15]	1.21
D	O	3	[0.81 0.94 1.00 1.00 0.85 0.96]	[0.31 0.26 0.09 **0.02** 0.03 0.32 0.26 0.10 0.21]	[0. 0. 0. 1. 0. 0. 0. 0. 0.]	[-0.31 -0.26 -0.09 **0.98** -0.03 -0.32 -0.26 -0.10 -0.21]	1.35
E	E	4	[0.99 1.00 1.00 1.00 0.49 0.92]	[0.33 0.27 0.12 0.02 **0.02** 0.34 0.28 0.12 0.13]	[0. 0. 0. 0. 1. 0. 0. 0. 0.]	[-0.33 -0.27 -0.12 -0.02 **0.98** -0.34 -0.28 -0.12 -0.13]	1.38
I	I	5	[1.00 1.00 0.99 0.52 1.00 0.99]	[0.35 0.28 0.11 0.02 0.04 **0.33** 0.40 0.06 0.28]	[0. 0. 0. 0. 0. 1. 0. 0. 0.]	[-0.35 -0.28 -0.11 -0.02 -0.04 **0.67** -0.40 -0.06 -0.28]	0.90
K	K	6	[0.72 1.00 0.69 0.99 0.42 1.00]	[0.23 0.21 0.14 0.04 0.03 0.39 **0.23** 0.19 0.19]	[0. 0. 0. 0. 0. 0. 1. 0. 0.]	[-0.24 -0.21 -0.14 -0.04 -0.03 -0.39 **0.77** -0.19 -0.19]	0.94
L	L	7	[0.90 1.00 0.81 0.97 0.39 0.98]	[0.28 0.24 0.15 0.03 0.03 0.40 0.26 **0.16** 0.16]	[0. 0. 0. 0. 0. 0. 0. 1. 0.]	[-0.28 -0.24 -0.15 -0.03 -0.03 -0.40 -0.26 **0.84** -0.16]	1.11
M	M	8	[0.10 1.00 0.90 1.00 0.98 1.00]	[0.23 0.22 0.07 0.04 0.07 0.22 0.18 0.14 **0.38**]	[0. 0. 0. 0. 0. 0. 0. 0. 1.]	[-0.23 -0.22 -0.07 -0.04 -0.07 -0.22 -0.18 -0.14 **0.62**]	0.60

Table 9

Alpha = 1.5 Eta = 0.5 maxNumIterations = 10000 Epsilon = 0.3

Input Letter (Third Element)	Desired Output (Ffith Element)	Desired Output Corresponding Number (Fourth Element)	Hidden Node Activations	Output Node Activations	Desired Output Array Values	Error Values	New SSE
A	A	0	[0.94 0.16 0.23 0.29 0.02 0.00]	[**0.93** 0.86 0.18 0.72 0.16 0.60 0.96 0.76 0.57]	[1. 0. 0. 0. 0. 0. 0. 0. 0.]	[**0.07** -0.86 -0.18 -0.73 -0.16 -0.60 -1.00 -0.76 -0.57]	3.51
B	B	1	[0.01 0.00 0.00 0.11 0.08 0.99]	[0.87 **0.71** 0.20 0.61 0.41 0.86 0.91 0.86 0.79]	[0. 1. 0. 0. 0. 0. 0. 0. 0.]	[-0.87 **0.29** -0.20 -0.61 -0.41 -0.86 -0.91 -0.85 -0.79]	4.32
C	C	2	[0.83 0.01 0.00 0.57 0.60 0.93]	[0.93 0.67 **0.07** 0.39 0.27 0.77 0.98 0.73 0.68]	[0. 0. 1. 0. 0. 0. 0. 0. 0.]	[-0.93 -0.67 **0.93** -0.39 -0.27 -0.77 -0.98 -0.73 -0.68]	4.95
D	O	3	[1.00 0.00 0.00 0.03 0.05 1.00]	[0.94 0.85 0.14 **0.66** 0.16 0.77 0.98 0.85 0.81]	[0. 0. 0. 1. 0. 0. 0. 0. 0.]	[-0.94 -0.85 -0.14 **0.34** -0.16 -0.77 -0.98 -0.85 -0.81]	4.70
E	E	4	[0.43 0.00 0.00 0.93 0.06 0.29]	[0.92 0.70 0.09 0.62 **0.47** 0.79 0.90 0.79 0.43]	[0. 0. 0. 0. 1. 0. 0. 0. 0.]	[-0.92 -0.70 -0.09 -0.62 **0.53** -0.79 -0.90 -0.79 -0.43]	4.24
I	I	5	[0.04 0.03 0.57 0.64 0.78 0.32]	[0.83 0.56 0.10 0.30 0.50 **0.69** 0.96 0.49 0.53]	[0. 0. 0. 0. 0. 1. 0. 0. 0.]	[-0.83 -0.56 -0.10 -0.30 -0.50 **0.31** -0.96 -0.49 -0.53]	2.90
K	K	6	[0.15 0.24 0.03 0.12 0.91 1.00]	[0.87 0.49 0.22 0.31 0.26 0.79 **0.97** 0.71 0.78]	[0. 0. 0. 0. 0. 0. 1. 0. 0.]	[-0.87 -0.49 -0.22 -0.31 -0.26 -0.79 **0.03** -0.71 -0.78]	2.95
L	L	7	[0.44 0.16 0.00 0.16 0.86 0.96]	[0.89 0.57 0.17 0.34 0.22 0.75 0.98 **0.71** 0.78]	[0. 0. 0. 0. 0. 0. 0. 1. 0.]	[-0.89 -0.57 -0.17 -0.34 -0.22 -0.75 -0.98 **0.30** -0.78]	3.53
M	M	8	[0.27 0.99 0.01 0.02 0.02 0.07]	[0.94 0.67 0.65 0.72 0.10 0.83 0.93 0.89 **0.49**]	[0. 0. 0. 0. 0. 0. 0. 0. 1.]	[-0.94 -0.67 -0.65 -0.72 -0.10 -0.83 -0.93 -0.89 **0.51**]	4.87

Table 10

Alpha = 1.5 Eta = 0.5 maxNumIterations = 100 Epsilon = 0.3

Input Letter (Third Element)	Desired Output (Ffith Element)	Desired Output Corresponding Number (Fourth Element)	Hidden Node Activations	Output Node Activations	Desired Output Array Values	Error Values	New SSE
A	A	0	[0.99 1.00 0.99 0.43 0.17 0.62]	[**0.02** 0.80 0.39 0.51 0.50 0.12 0.93 0.83 0.19]	[1. 0. 0. 0. 0. 0. 0. 0. 0.]	[**0.98** -0.80 -0.39 -0.51 -0.50 -0.12 -0.93 -0.83 -0.19]	3.86
B	B	1	[0.94 1.00 1.00 0.20 0.71 0.99]	[0.04 **0.85** 0.47 0.43 0.25 0.27 0.94 0.73 0.22]	[0. 1. 0. 0. 0. 0. 0. 0. 0.]	[-0.04 **0.15** -0.47 -0.43 -0.25 -0.27 -0.94 -0.73 -0.22]	2.03
C	C	2	[0.95 1.00 1.00 0.02 0.89 0.77]	[0.04 0.90 **0.42** 0.36 0.27 0.26 0.92 0.81 0.23]	[0. 0. 1. 0. 0. 0. 0. 0. 0.]	[-0.04 -0.90 **0.59** -0.36 -0.27 -0.26 -0.92 -0.81 -0.23]	2.99
D	O	3	[0.82 1.00 1.00 0.01 0.11 0.79]	[0.04 0.79 0.46 **0.46** 0.35 0.12 0.96 0.87 0.18]	[0. 0. 0. 1. 0. 0. 0. 0. 0.]	[-0.04 -0.79 -0.46 **0.54** -0.35 -0.12 -0.96 -0.87 -0.18]	2.97
E	E	4	[0.84 1.00 0.96 0.08 0.92 0.93]	[0.05 0.89 0.48 0.38 **0.20** 0.29 0.92 0.74 0.26]	[0. 0. 0. 0. 1. 0. 0. 0. 0.]	[-0.05 -0.89 -0.48 -0.38 **0.80** -0.29 -0.92 -0.74 -0.26]	3.36
I	I	5	[0.99 0.98 1.00 0.15 0.73 0.99]	[0.04 0.86 0.45 0.41 0.27 **0.30** 0.94 0.75 0.21]	[0. 0. 0. 0. 0. 1. 0. 0. 0.]	[-0.04 -0.86 -0.45 -0.41 -0.27 **0.70** -0.94 -0.75 -0.21]	3.16
K	K	6	[0.68 1.00 0.39 1.00 0.06 0.35]	[0.05 0.70 0.55 0.46 0.40 0.10 **0.72** 0.71 0.43]	[0. 0. 0. 0. 0. 0. 1. 0. 0.]	[-0.05 -0.70 -0.55 -0.46 -0.40 -0.10 **0.28** -0.71 -0.43]	1.94
L	L	7	[0.96 1.00 0.93 0.35 0.25 0.69]	[0.03 0.80 0.42 0.47 0.42 0.14 0.93 **0.82** 0.21]	[0. 0. 0. 0. 0. 0. 0. 1. 0.]	[-0.03 -0.80 -0.42 -0.47 -0.42 -0.14 -0.93 **0.18** -0.21]	2.17
M	M	8	[0.01 0.97 1.00 1.00 0.96 0.09]	[0.06 0.95 0.46 0.58 0.32 0.04 0.62 0.60 **0.48**]	[0. 0. 0. 0. 0. 0. 0. 0. 1.]	[-0.06 -0.95 -0.50 -0.58 -0.32 -0.04 -0.62 -0.60 **0.52**]	2.58

Table 11

Alpha = 1.5 Eta = 1.0 maxNumIterations = 1000 Epsilon = 0.3

Input Letter (Third Element)	Desired Output (Ffith Element)	Desired Output Corresponding Number (Fourth Element)	Hidden Node Activations	Output Node Activations	Desired Output Array Values	Error Values	New SSE
A	A	0	[0.04 0.54 0.05 0.05 1.00 0.26]	[0.50 0.55 0.38 0.85 0.05 0.10 0.74 0.55 0.23]	[1. 0. 0. 0. 0. 0. 0. 0. 0.]	[0.50 -0.55 -0.38 -0.85 -0.05 -0.10 -0.74 -0.55 -0.23]	2.33
B	B	1	[0.15 0.01 0.84 0.02 0.99 0.00]	[0.33 0.46 0.18 0.58 0.15 0.04 0.70 0.22 0.31]	[0. 1. 0. 0. 0. 0. 0. 0. 0.]	[-0.33 0.54 -0.18 -0.58 -0.15 -0.04 -0.70 -0.22 -0.31]	1.44
C	C	2	[0.57 0.70 0.84 0.00 0.93 0.00]	[0.38 0.67 0.26 0.47 0.14 0.05 0.88 0.34 0.20]	[0. 0. 1. 0. 0. 0. 0. 0. 0.]	[-0.38 -0.67 0.74 -0.47 -0.14 -0.05 -0.88 -0.34 -0.20]	2.29
D	O	3	[0.35 0.40 0.98 0.24 1.00 0.00]	[0.36 0.47 0.20 0.44 0.17 0.05 0.76 0.26 0.24]	[0. 0. 0. 1. 0. 0. 0. 0. 0.]	[-0.36 -0.47 -0.20 0.56 -0.17 -0.05 -0.76 -0.26 -0.24]	1.43
E	E	4	[0.11 0.03 0.30 0.00 0.95 0.00]	[0.36 0.50 0.28 0.74 0.11 0.06 0.68 0.33 0.29]	[0. 0. 0. 0. 1. 0. 0. 0. 0.]	[-0.36 -0.50 -0.28 -0.74 0.89 -0.06 -0.68 -0.33 -0.29]	2.46
I	I	5	[0.02 0.05 0.95 0.00 0.46 0.12]	[0.40 0.56 0.19 0.48 0.20 0.07 0.63 0.32 0.30]	[0. 0. 0. 0. 0. 1. 0. 0. 0.]	[-0.40 -0.5 -06.19 -0.48 -0.20 0.93 -0.63 -0.32 -0.30]	2.22
K	K	6	[0.95 0.54 0.00 0.00 0.53 0.07]	[0.35 0.78 0.53 0.52 0.24 0.09 0.89 0.49 0.28]	[0. 0. 0. 0. 0. 0. 1. 0. 0.]	[-0.35 -0.78 -0.53 -0.52 -0.24 -0.09 0.11 -0.49 -0.28]	1.68
L	L	7	[0.47 0.35 0.04 0.00 1.00 0.01]	[0.36 0.61 0.40 0.74 0.11 0.07 0.82 0.41 0.26]	[0. 0. 0. 0. 0. 0. 0. 1. 0.]	[-0.36 -0.61 -0.40 -0.74 -0.11 -0.07 -0.82 0.59 -0.26]	2.31
M	M	8	[0.97 0.98 0.01 0.01 0.97 0.00]	[0.39 0.77 0.53 0.63 0.12 0.07 0.93 0.53 0.18]	[0. 0. 0. 0. 0. 0. 0. 0. 1.]	[-0.39 -0.77 -0.53 -0.63 -0.12 -0.07 -0.93 -0.53 0.82]	3.27

Analysis and Interpretation

Results show that changing parameters impact the Summed Squared Error (SSE) considering that the desired outputs are set so each input letter (Element 3) is given a desired output (Element 5). First, all parameters were kept consistent while Epsilon, the criteria for accepting that the network has been sufficiently trained, was changed. When keeping Alpha, the steepness of the transfer function, to 1.0, Eta, the learning rate, to 0.5, and maxNumIterations, total number of iterations, to 5000, Epsilon, was first set to 0.01. The results showed that the error values corresponded with the desired output node so that the position of the desired output was positive while the remaining eight values were negative, but at this stage, the SSE did not provide a clear idea of the impact of Epsilon [see Table 1 in Results]. As the Epsilon was increased from 0.01 to 0.1, the SSE seemed to increase for every data set. This trend continued when increasing the Epsilon to 0.2 and 0.3 [see Tables 2, 3, and 4 in Results].

After setting the Epsilon to 0.3 moving forward, the Alpha was reduced to 0.5, which generally resulted in additional decrease of the SSE for the data sets [see Table 5 in Results]. When increasing the Alpha to 1.5, there was generally a greater decrease in the SSE for the data sets, so the Alpha was further increased to 2.0, but the SSE appeared to increase [see Tables 6 and 7 in Results]. Therefore, the Alpha was permanently set to 1.5 moving forward. When the maxNumIterations parameter was reduced from 5,000 to 1,000, the SSE for inputs fell below 1.5 and even went below 1.00 for several inputs. This experiment had the lowest overall SSE [see Table 8 in Results]. However, increasing the maxNumIterations to 10,000 caused the SSE for inputs to increase extensively. Decreasing the maxNumIterations parameter to 100 brought the SSE down for inputs, but still fairly high compared to when it was set to 1,000 [see Tables 9 and 10 in Results]. When setting the Alpha to 1.5, maxNumIterations to 1,000, and Epsilon to 0.3, Eta was increased from 0.5 to 1.0, but the SSE values increased compared to the results when Eta was 0.5 [see Table 11 in Results].

When considering the output node activations and error values, there appeared to be a relationship between the parameters and the SSE of the inputs. As the SSE decreased, the error values increased towards one based on the desired output. For example, input letter “A,” which has a desired output array of [1. 0. 0. 0. 0. 0. 0. 0. 0.] had error values of [0.83 -0.13 -0.22 -0.06 -0.03 -0.61 -0.23 -0.31 -0.15] when the Alpha was 1.5, Eta was 0.5, maxNumIterations was 1,000, and Epsilon was 0.3 [see Table 8 in Results]. However, the error values were [0.18 -0.55 -0.16 -0.59 -0.47 -0.60 -0.58 -0.32 -0.46] when the Alpha is 1.0, Eta was 0.5, maxNumIterations is 5,000 and Epsilon is 0.2 [see Table 3].

Ultimately, every parameter tested played a key role in reducing the data sets’ SSE, but the biggest results came from increasing the Epsilon and decreasing the maxNumIterations to an ideal point. With every parameter tested in this research, increasing or decreasing the value too far would end up increasing the SSE values after those values were previously decreased. Through several experiments, a “sweet spot” can certainly be found. In this research, that “sweet spot” appears to be when Alpha is 1.5, Eta is 0.5, maxNumIterations is 1,000, and Epsilon is 0.3.

Conclusions

The experiments ran in this research showed that it can still be difficult to tell how inner workings of a model take place but using a gray-box model still provides solid control in order to determine desired results. Such a gray-box model would be recommended for studies that require classification with accuracy and control so the results are as expected. It can still be difficult to obtain a balance between interpretability and accuracy through gray-box modeling but understanding how to finetune the parameters makes this very possible. Essentially, when comparing to black-box and white-box models, gray-box models can be used to create a process that provides control of inner workings and desired accuracy.

Directions for Future Work

This research can be taken in several direction in the future. Firstly, the parameters can be further investigated so that increasing and decreasing many times can help to come to an ideal point. Ultimately, experimenting with an extensive number of combinations can provide a far deeper understanding of the inner workings. Additionally, more training sets can be added and new “big shape classes” can be created, and while this may create more variability in results, it can help to gain understanding of the role the data itself can play rather than just the parameters of the model. Along with new variants, random noise can also be added to improve robustness of the network, which can lead to better results. Finally, diving into the connections between the input layers and hidden layers, as well as the hidden layers and outputs should be done in the future. While it is clear that backpropagation is used in the code, a proper dive into the code is necessary to understand how weights are being utilized, how node activations are derived, and how error values are calculated.

Code and Data Availability

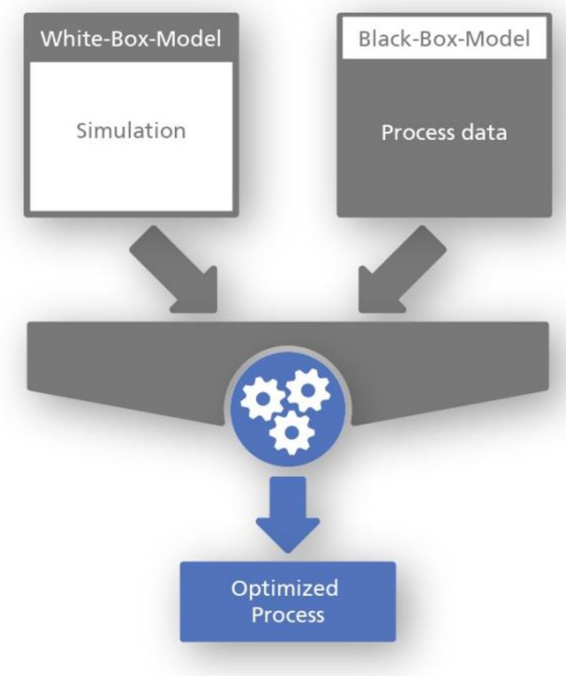
I am truly grateful to Dr. Alianna Maren and Robert Guenther for sharing the code and data used in this study.

Acknowledgements

I would like to express my gratitude to Dr. Alianna Maren and Robert Guenther for providing a breakdown of the neural network used in this study, including detailed insight into the training data sets and visual diagrams. I would also like to thank Zubair Ahmed for peer reviewing

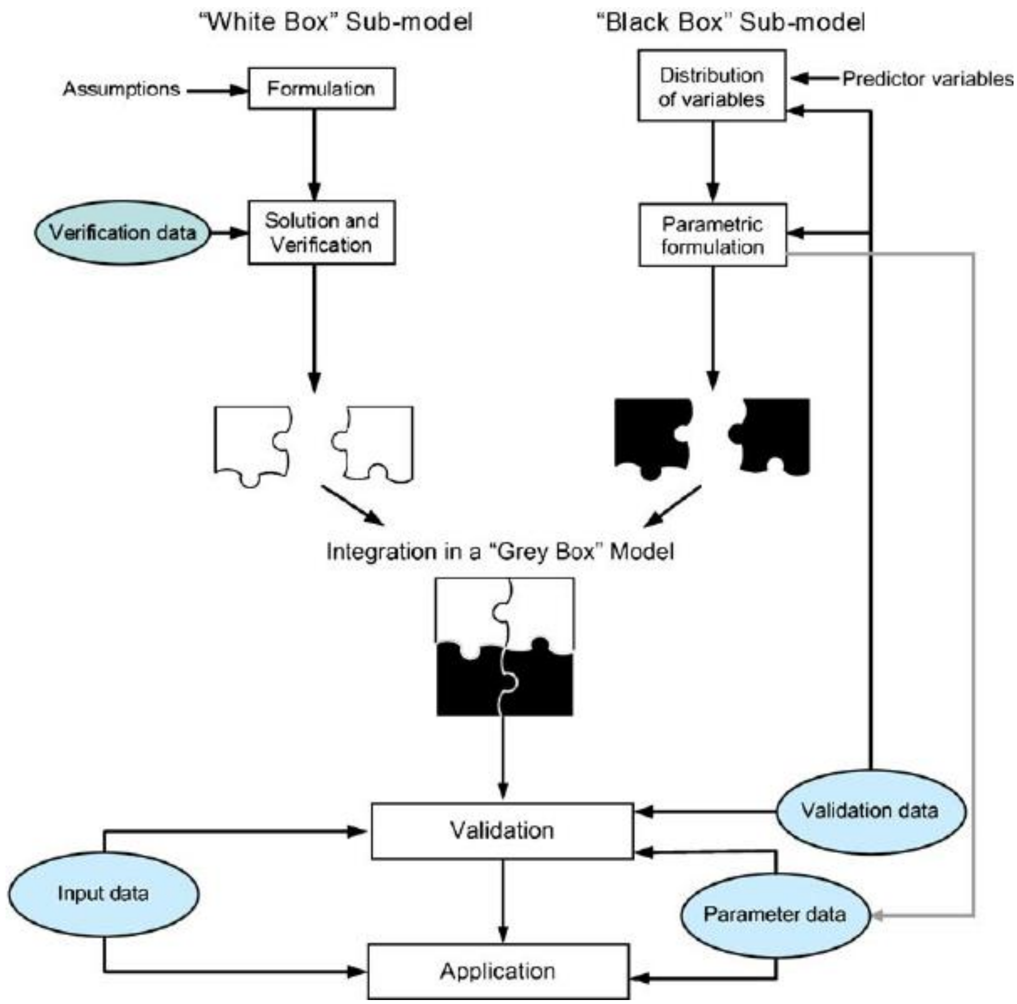
Appendix

Figure 1: Schematic graphic grey-box model



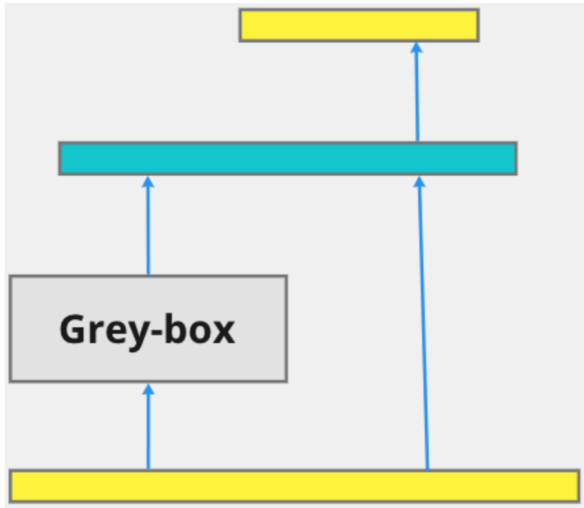
This model highlights the basic idea of an optimized process obtained through the combination of a white-box model and a black-box model (Bortz n.d.)

Figure 2: Flow diagram of models



This diagram illustrates the development processes and data requirements of white-box, black-box, and grey-box submodels in the study (Estrada-Flores et al. 2006).

Figure 3: Complex Neural Network with grey-box big shape pre-classifier



The grey-box in this network is a “get-a-clue” component. The output from this grey-box is a large-scale “letter-type classification (one of N classes, N < 26). The hidden layer gets input from two sources, including the regular 81-elements input nodes and the outputs from the grey-box. The grey-box is completely trained before added to the new big neural network, so the pretrained grey-box gives an overall classification that supports a detailed classification.

Figure 4: 81-element input grid training data list

trainingDataListA0 = (1,[0,0,0,0,1,0,0,0,0, 0,0,0,1,0,1,0,0,0, 0,0,1,0,0,0,1,0,0, 0,1,0,0,0,0,0,1,0, 1,1,1,1,1,1,1,1,1, 1,0,0,0,0,0,0,0,1, 1,0,0,0,0,0,0,0,1, 1,0,0,0,0,0,0,0,1, 1,0,0,0,0,0,0,0,1],1,'A',1,'A') # training data list 1 selected for the letter 'A'

Element 1
Element 2
Element 3
Element 4
Element 5

Resources

Barkman, Patrick. 2018. *Grey-box modelling of distributed parameter systems*. Stockhold:

Bortz, Michael. n.d. "Grey Box Model – Integrating Application Knowledge in the Learning Process." Accessed April 20, 2022. https://www.itwm.fraunhofer.de/en/departments/opt/machine-learning-hybrid-models/grey-box-model-integrating-application-knowledge-learning-process.html#:~:text=Compared%20to%20the%20black%20box,information%20available%20from%20existing%20data.

Daniel, Lowrie. 2021. "What is White vs. Grey vs. Black Box Testing?" Filmed 2021 at ITProTV. https://www.youtube.com/watch?v=CPVMbZYeiF4.

Estrada-Flores, Silvia, Bart De Ketelaere, I. Merts, and Jeroen Lammertyn. 2006. "Development and validation of 'grey-box' models for refrigeration applications: A review of key concepts." International Journal of Refrigeration 29(6): 931-946. 10.1016/j.ijrefrig.2006.03.018.

Pintelas, Emmanuel, Ioannis E. Livieris, and Panagiotis Pintelas. 2020. "A Grey-Box Ensemble Model Exploiting Black-Box Accuracy and White-Box Intrinsic Interpretability." *Algorithms*. 13(1). https://doi.org/10.3390/a13010017.

Sohlberg, Bjorn. 1998. *Supervision and Control for Industrial Processes*, 7-43. Borlange: Springer.

Walker, Chris. 2020. "White Box vs Black Box Models: Balancing Interpretability and Accuray." Accessed May 02, 2022. https://blog.dataiku.com/white-box-vs-black-box-models-balancing-interpretability-and-accuracy.

Santiago Zanella-Beguelin, Shruti Tople, Andrew Paverd, and Boris Köpf. 2021. Proceedings of the 38th International Conference on Machine Learning, PMLR 139:12278-12286. https://proceedings.mlr.press/v139/zanella-beguelin21a.html.