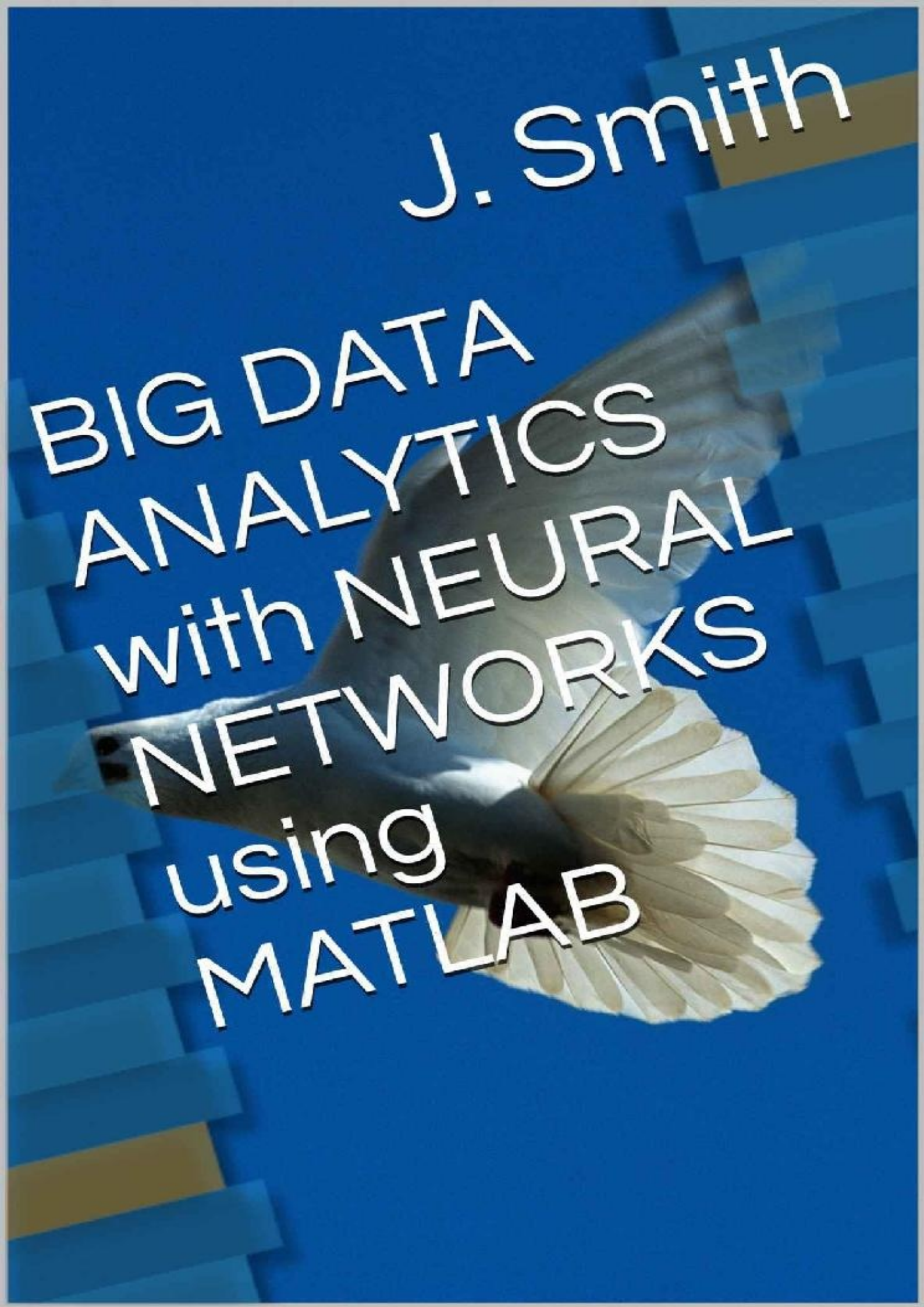
J. Smith

BIG DATA ANALYTICS with NEURAL NETWORKS using MATLAB



BIG DATA ANALYTICS WITH NEURAL NETWORKS USING MATLAB

J. SMITH

CONTENTS

Chapter 1

INTRODUCTION TO BIGADATA ANALYTICS AND NEURAL NETWORKS WITH MATLAB

1.1 BIG DATA ANALYTICS

Big data analytics examines large amounts of data to uncover hidden patterns, correlations and other insights. With today's technology, it's possible to analyze your data and get answers from it almost immediately – an effort that's slower and less efficient with more traditional business intelligence solutions.

The concept of big data has been around for years; most organizations now understand that if they capture all the data that streams into their businesses, they can apply analytics and get significant value from it. But even in the 1950s, decades before anyone uttered the term "big data," businesses were using basic analytics (essentially numbers in a spreadsheet that were manually examined) to uncover insights and trends.

Big data analytics helps organizations harness their data and use it to identify new opportunities. That, in turn, leads to smarter business moves, more efficient operations, higher profits and happier customers. In his report *Big Data in Big Companies*, IIA Director of Research Tom Davenport interviewed more than 50 businesses to understand how they used big data. He found they got value in the following ways:

- **Cost reduction.** Big data technologies such as Hadoop and cloud-based analytics bring significant cost advantages when it comes to storing large amounts of data – plus they can identify more efficient ways of doing business.
- **Faster, better decision making.** With the speed of Hadoop and in-memory analytics, combined with the ability to analyze new sources of data, businesses are able to analyze information immediately – and make decisions based on what they've learned.
- **New products and services.** With the ability to gauge customer needs and satisfaction through analytics comes the power to give customers what they want. Davenport points out that with big data analytics, more companies are creating new products to meet customers' needs.

The new benefits that big data analytics brings to the table, however, are speed and efficiency. Whereas a few years ago a business would have gathered information, run analytics and unearthed information that could be used for future decisions, today that business can identify insights for immediate decisions. The ability to work faster – and stay agile – gives organizations a competitive edge they didn't have before

There's no single technology that encompasses big data analytics. Of course, there's advanced analytics that can be applied to big data, but in reality several types of technology work together to help you get the most value from your information. Here are the biggest players:

Data management. Data needs to be high quality and well-governed before it can be reliably analyzed. With data constantly flowing in and out of an organization, it's important to establish repeatable processes to build and maintain standards for data

quality. Once data is reliable, organizations should establish a master data management program that gets the entire enterprise on the same page.

Data mining. Data mining technology helps you examine large amounts of data to discover patterns in the data – and this information can be used for further analysis to help answer complex business questions. With data mining software, you can sift through all the chaotic and repetitive noise in data, pinpoint what's relevant, use that information to assess likely outcomes, and then accelerate the pace of making informed decisions.

Hadoop. This open source software framework can store large amounts of data and run applications on clusters of commodity hardware. It has become a key technology to doing business due to the constant increase of data volumes and varieties, and its distributed computing model processes big data fast. An additional benefit is that Hadoop's open source framework is free and uses commodity hardware to store large quantities of data.

In-memory analytics. By analyzing data from system memory (instead of from your hard disk drive), you can derive immediate insights from your data and act on them quickly. This technology is able to remove data prep and analytical processing latencies to test new scenarios and create models; it's not only an easy way for organizations to stay agile and make better business decisions, it also enables them to run iterative and interactive analytics scenarios.

Predictive analytics. Predictive analytics technology uses data, statistical algorithms and machine-learning techniques to identify the likelihood of future outcomes based on historical data. It's all about providing a best assessment on what will happen in the future, so organizations can feel more confident that they're making the best possible business decision. Some of the most common applications of predictive analytics include fraud detection, risk, operations and marketing.

Text mining. With text mining technology, you can analyze text data from the web, comment fields, books and other text-based sources to uncover insights you hadn't noticed before. Text mining uses machine learning or natural language processing technology to comb through documents – emails, blogs, Twitter feeds, surveys, competitive intelligence and more – to help you analyze large amounts of information and discover new topics and term relationships.

1.2 BIG DATA ANALYTICS AND MATLAB

Big data analytics is the process of collecting, organizing and analyzing large sets of data (*called* big data) to discover patterns and other useful information. Big data analytics can help organizations to better understand the information contained within the data and will also help identify the data that is most important to the business and future business decisions. Analysts working with big data basically want the *knowledge* that comes from analyzing the data.

To analyze such a large volume of data, big data analytics is typically performed using specialized software tools and applications for predictive analytics, data mining, text mining, forecasting and data optimization. Collectively these processes are separate but highly integrated functions of high-performance analytics. Using big data tools and software enables an organization to process extremely large volumes of data that a business has collected to determine which data is relevant and can be analyzed to drive better business decisions in the future. Among all these tools highlights MATLAB.

MATLAB implements various toolboxes for working on big data analytics, such as Statistics Toolbox and Neural Network Toolbox.

1.3 PREDICTIVE MODELS AND DATA ANALTICS. NEURAL NETWORKS

Predictive models and analysis are typically used to forecast future probabilities. Applied to business, predictive models are used to analyze current data and historical facts in order to better understand customers, products and partners and to identify potential risks and opportunities for a company. It uses a number of techniques, including data mining, statistical modeling and machine learning to help analysts make future business forecasts.

Among the most interesting predictive models are the neural networks. MATLAB implementrs the Neural Network Toolbox specialized in the techniques of analytics based on neural networks. Throughout this book the techniques of analytics based on neural networks are developed using MATLAB software and in concrete Neural Network Toolbox. The Neural Network Toolbox software uses the network object to store all of the information that defines a neural network.

1.4 NEURAL NETWORK TOOLBOX

MATLAB has the tool Neural Network Toolbox that provides algorithms, functions, and apps to create, train, visualize, and simulate neural networks. You can perform classification, regression, clustering, dimensionality reduction, time-series forecasting, and dynamic system modeling and control.

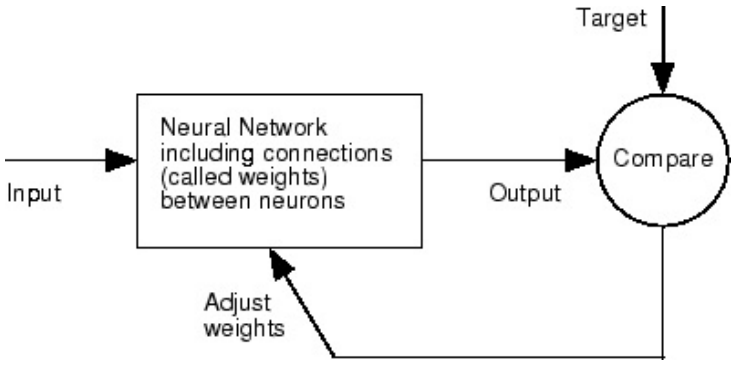
The toolbox includes convolutional neural network and autoencoder deep learning algorithms for image classification and feature learning tasks. To speed up training of large data sets, you can distribute computations and data across multicore processors, GPUs, and computer clusters using Parallel Computing Toolbox.

The more important features are the following:

- Deep learning, including convolutional neural networks and autoencoders
- Parallel computing and GPU support for accelerating training (with Parallel Computing Toolbox)
- Supervised learning algorithms, including multilayer, radial basis, learning vector quantization (LVQ), time-delay, nonlinear autoregressive (NARX), and recurrent neural network (RNN)
- Unsupervised learning algorithms, including self-organizing maps and competitive layers
- Apps for data-fitting, pattern recognition, and clustering
- Preprocessing, postprocessing, and network visualization for improving training efficiency and assessing network performance
- Simulink® blocks for building and evaluating neural networks and for control systems applications

Neural networks are composed of simple elements operating in parallel. These elements are inspired by biological nervous systems. As in nature, the connections between elements largely determine the network function. You can train a neural network to perform a particular function by adjusting the values of the connections (weights) between elements.

Typically, neural networks are adjusted, or trained, so that a particular input leads to a specific target output. The next figure illustrates such a situation. Here, the network is adjusted, based on a comparison of the output and the target, until the network output matches the target. Typically, many such input/target pairs are needed to train a network.



Neural networks have been trained to perform complex functions in various fields, including pattern recognition, identification, classification, speech, vision, and control systems.

Neural networks can also be trained to solve problems that are difficult for conventional computers or human beings. The toolbox emphasizes the use of neural network paradigms that build up to—or are themselves used in— engineering, financial, and other practical applications.

1.5 USING NEURAL NETWORK TOOLBOX

There are four ways you can use the Neural Network Toolbox software.

- The first way is through its tools. You can open any of these tools from a master tool started by the command nnstart. These tools provide a convenient way to access the capabilities of the toolbox for the following tasks:
  - Function fitting (*nftool*)
  - Pattern recognition (*nprtool*)
  - Data clustering (*nctool*)
  - Time-series analysis (*ntstool*)
- The second way to use the toolbox is through basic command-line operations.

The command-line operations offer more flexibility than the tools, but with some added complexity. If this is your first experience with the toolbox, the tools provide the best introduction. In addition, the tools can generate scripts of documented MATLAB code to provide you with templates for creating your own customized command-line functions. The process of using the tools first, and then generating and modifying MATLAB scripts, is an excellent way to learn about the functionality of the toolbox.

- The third way to use the toolbox is through customization. This advanced capability allows you to create your own custom neural networks, while still having access to the full functionality of the toolbox. You can create networks with arbitrary connections, and you still be able to train them using existing toolbox training functions (as long as the network components are differentiable).

- The fourth way to use the toolbox is through the ability to modify any of the functions contained in the toolbox. Every computational component is written in MATLAB code and is fully accessible.

These four levels of toolbox usage span the novice to the expert: simple tools guide the new user through specific applications, and network customization allows researchers to try novel architectures with minimal effort. Whatever your level of neural network and MATLAB knowledge, there are toolbox features to suit your needs.

1.6 AUTOMATIC SCRIPT GENERATION

The tools themselves form an important part of the learning process for the Neural Network Toolbox software. They guide you through the process of designing neural networks to solve problems in four important application areas, without requiring any background in neural networks or sophistication in using MATLAB. In addition, the tools can automatically generate both simple and advanced MATLAB scripts that can reproduce the steps performed by the tool, but with the option to override default settings. These scripts can provide you with templates for creating customized code, and they can aid you in becoming familiar with the command-line functionality of the toolbox. It is highly recommended that you use the automatic script generation facility of these tools.

Chapter 2

FIT DATA WITH A NEURAL NETWORK. GRAPHICAL INTERFACE

2.1 INTRODUCTION

Neural networks are good at fitting functions. In fact, there is proof that a fairly simple neural network can fit any practical function.

Suppose, for instance, that you have data from a housing application. You want to design a network that can predict the value of a house (in $1000s), given 13 pieces of geographical and real estate information. You have a total of 506 example homes for which you have those 13 items of data and their associated market values.

You can solve this problem in two ways:

- Use a graphical user interface, *nftool*, as described in Using the Neural Network Fitting Tool.
- Use command-line functions, as described in Using Command-Line Functions.

It is generally best to start with the GUI, and then to use the GUI to automatically generate command-line scripts. Before using either method, first define the problem by selecting a data set. Each GUI has access to many sample data sets that you can use to experiment with the toolbox (see Neural Network Toolbox Sample Data Sets). If you have a specific problem that you want to solve, you can load your own data into the workspace. The next section describes the data format.

To define a fitting problem for the toolbox, arrange a set of Q input vectors as columns in a matrix. Then, arrange another set of Q target vectors (the correct output vectors for each of the input vectors) into a second matrix (see "Data Structures" for a detailed description of data formatting for static and time-series data). For example, you can define the fitting problem for a Boolean AND gate with four sets of two-element input vectors and one-element targets as follows:

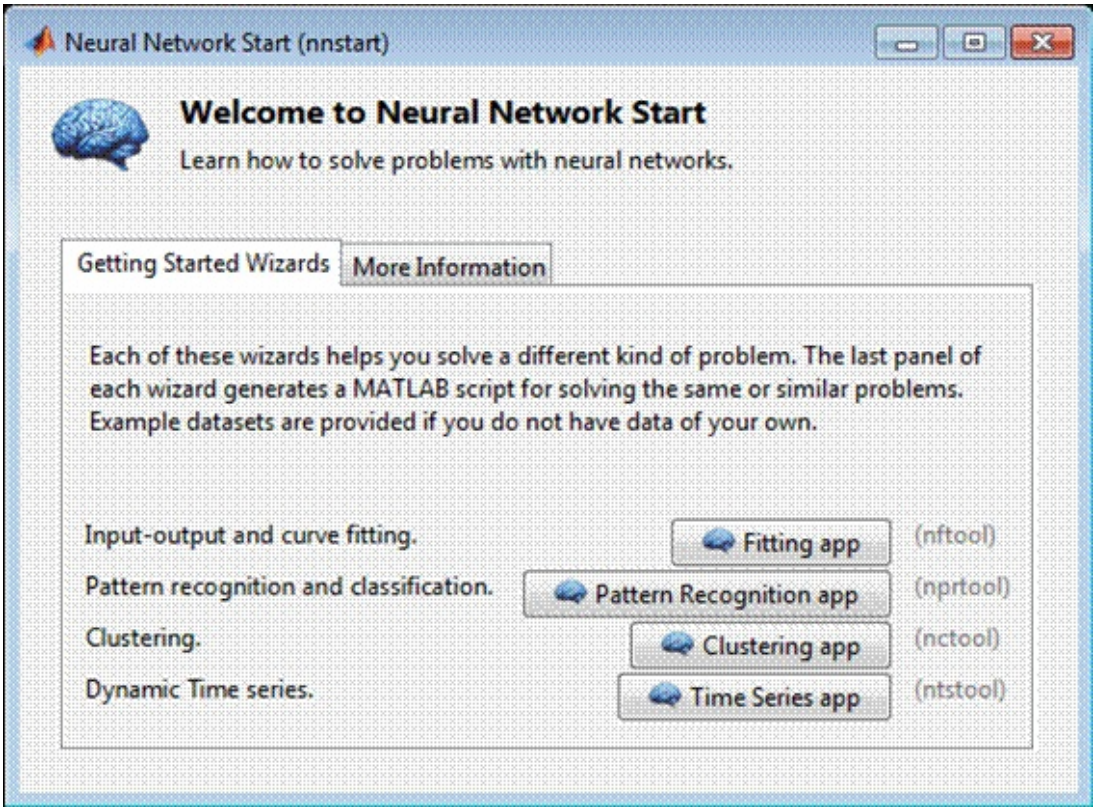
```
inputs = [0 1 0 1; 0 0 1 1];

targets = [0 0 0 1];
```

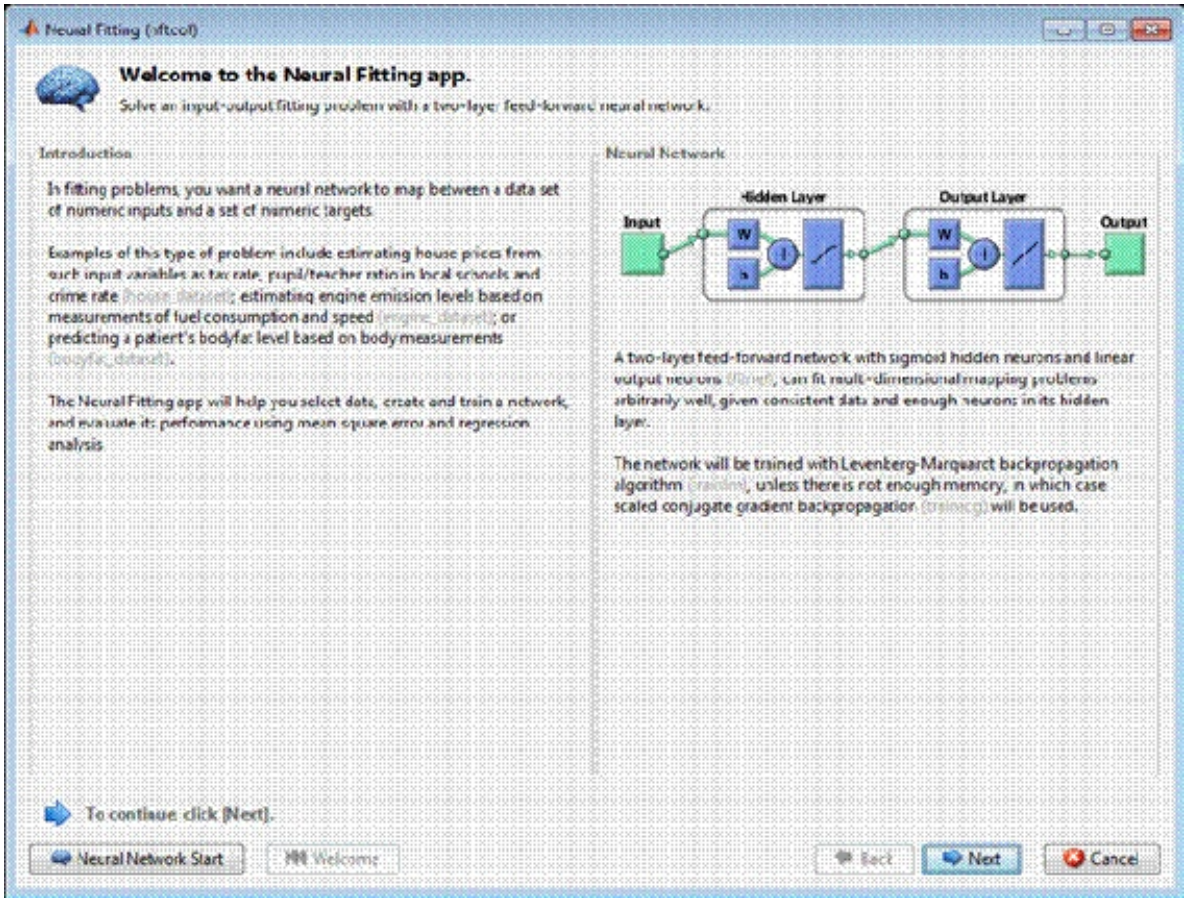
The next section shows how to train a network to fit a data set, using the neural network fitting tool GUI, *nftool*. This example uses the housing data set provided with the toolbox.

2.2 USING THE NEURAL NETWORK FITTING TOOL

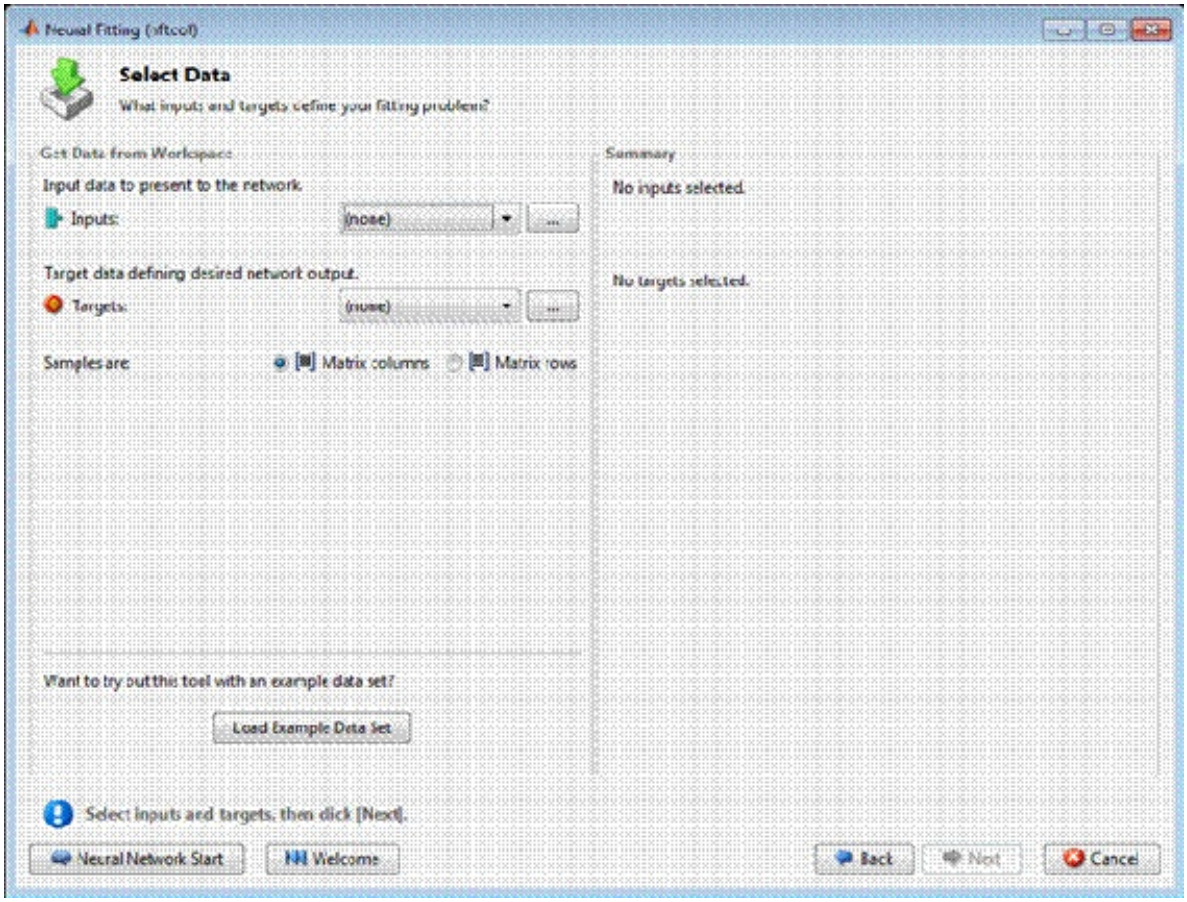
· Open the Neural Network Start GUI with this command: *nnstart*



· Click *Fitting Tool* to open the Neural Network Fitting Tool. (You can also use the command *nftool*.)



· Click *Nex*t to proceed.



· Click *Load Example Data Set* in the Select Data window. The Fitting Data Set Chooser window opens.

Note Use the **Inputs** and **Targets** options in the Select Data window when you need to load data from the MATLAB workspace.