Introduction to Deep Learning Using R

A Step-by-Step Guide to Learning and Implementing Deep Learning Models Using R

Taweh Beysolow II

Apress®

Introduction to Deep Learning Using R

A Step-by-Step Guide to Learning and Implementing Deep Learning Models Using R

Taweh Beysolow II

Apress®

Introduction to Deep Learning Using R

Taweh Beysolow II
San Francisco, California, USA

ISBN-13 (pbk): 978-1-4842-2733-6 ISBN-13 (electronic): 978-1-4842-2734-3
DOI 10.1007/978-1-4842-2734-3

Library of Congress Control Number: 2017947908

Cover image designed by Freepik

Managing Director: Welmoed Spahr
Editorial Director: Todd Green
Acquisitions Editor: Celestin Suresh John
Development Editor: Laura Berendson
Technical Reviewer: Somil Asthana
Coordinating Editor: Sanchita Mandal
Copy Editor: Corbin Collins
Compositor: SPi Global
Indexer: SPi Global
Artist: SPi Global

Distributed to the book trade worldwide by Springer Science+Business Media New York, 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail orders-ny@springer-sbm.com, or visit www.springeronline.com. Apress Media, LLC is a California LLC and the sole member (owner) is Springer Science + Business Media Finance Inc (SSBM Finance Inc). SSBM Finance Inc is a **Delaware** corporation.

For information on translations, please e-mail rights@apress.com, or visit http://www.apress.com/rights-permissions.

Apress titles may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Print and eBook Bulk Sales web page at http://www.apress.com/bulk-sales.

Any source code or other supplementary material referenced by the author in this book is available to readers on GitHub via the book's product page, located at the following link:

https://github.com/TawehBeysolowII/AnIntroductionToDeepLearning.

For more detailed information, please visit http://www.apress.com/source-code.

Printed on acid-free paper

Contents at a Glance

Contents

About the Author

Taweh Beysolow II is a Machine Learning Scientist currently based in the United States with a passion for research and applying machine learning methods to solve problems. He has a Bachelor of Science degree in Economics from St. Johns University and a Master of Science in Applied Statistics from Fordham University. Currently, he is extremely passionate about all matters related to machine learning, data science, quantitative finance, and economics.

About the Technical Reviewer

Somil Asthana has a BTech from IITBHU India and an MS from the University of Buffalo, US, both in Computer Science. He is an Entrepreneur, Machine Learning Wizard, and BigData specialist consulting with fortune 500 companies like Sprint, Verizon, HPE, Avaya. He has a startup which provides BigData solutions and Data Strategies to Data Driven Industries in ecommerce, content / media domain.

Acknowledgments

To my family, who I am never grateful enough for. To my grandmother, from whom much was received and to whom much is owed. To my editors and other professionals who supported me through this process, no matter how small the assistance seemed. To my professors, who continue to inspire the curiosity that makes research worth pursuing. To my friends, new and old, who make life worth living and memories worth keeping. To my late friend Michael Giangrasso, who I intended on researching Deep Learning with. And finally, to my late mentor and friend Lawrence Sobol. I am forever grateful for your friendship and guidance, and continue to carry your teachings throughout my daily life.

Introduction

It is assumed that all readers have at least an elementary understanding of statistical or computer programming, specifically with respect to the R programming language. Those who do not will find it much more difficult to follow the sections of this book which give examples of code to use, and it is suggested that they return to this text upon gaining that information.

CHAPTER 1

Introduction to Deep Learning

With advances in hardware and the emergence of big data, more advanced computing methods have become increasingly popular. Increasing consumer demand for better products and companies seeking to leverage their resources more efficiently have also been leading this push. In response to these market forces, we have recently seen a renewed and widely spoken about interest in the field of machine learning. At the cross-section of statistics, mathematics, and computer science, *machine learning* refers to the science of creating and studying algorithms that improve their own behavior in an iterative manner by design. Originally, the field was devoted to developing artificial intelligence, but due to the limitations of the theory and technology that were present at the time, it became more logical to focus these algorithms on specific tasks. Most machine learning algorithms as they exist now focus on function optimization, and the solutions yielded don't always explain the underlying trends within the data nor give the inferential power that artificial intelligence was trying to get close to. As such, using machine learning algorithms often becomes a repetitive trial and error process, in which the choice of algorithm across problems yields different performance results. This is fine in some contexts, but in the case of language modeling and computer vision, it becomes problematic.

In response to some of the shortcomings of machine learning, and the significant advance in the theoretical and technological capabilities at our disposal today, deep learning has emerged and is rapidly expanding as one of the most exciting fields of science. It is being used in technologies such as self-driving cars, image recognition on social media platforms, and translation of text from one language to others. *Deep learning* is the subfield of machine learning that is devoted to building algorithms that explain and learn a high and low level of abstractions of data that traditional machine learning algorithms often cannot. The models in deep learning are often inspired by many sources of knowledge, such as game theory and neuroscience, and many of the models often mimic the basic structure of a human nervous system. As the field advances, many researchers envision a world where software isn't nearly as hard coded as it often needs to be today, allowing for a more robust, generalized solution to solving problems.

T. Beysolow II, *Introduction to Deep Learning Using R*, DOI 10.1007/978-1-4842-2734-3_1

Although it originally started in a space similar to machine learning, where the primary focus was constraint satisfaction to varying degrees of complexity, deep learning has now evolved to encompass a broader definition of algorithms that are able to understand multiple levels of representation of data that correspond to different hierarchies of complexity. In other words, the algorithms not only have predictive and classification ability, but they are able to learn different levels of complexity. An example of this is found in image recognition, where a neural network builds upon recognizing eyelashes, to faces, to people, and so on. The power in this is obvious: we can reach a level of complexity necessary to create intelligent software. We see this currently in features such as autocorrect, which models the suggested corrections to patterns of speech observed, specific to each person's vocabulary.

The structure of deep learning models often is such that they have *layers* of non-linear units that process data, or neurons, and the multiple layers in these models process different levels of abstraction of the data. Figure 1-1 shows a visualization of the layers of neural networks.

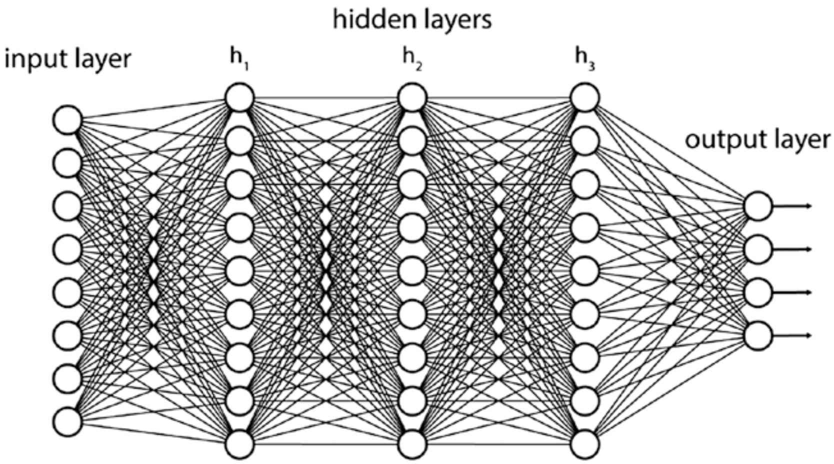

Figure 1-1. *Deep neural network*

Deep neural networks are distinguished by having many hidden layers, which are called "hidden" because we don't necessarily see what the inputs and outputs of these neurons are explicitly beyond knowing they are the output of the preceding layer. The addition of layers, and the functions inside the neurons of these layers, are what distinguish an individual architecture from another and establish the different use cases of a given model.

More specifically, lower levels of these models explain the "how," and the higher-levels of neural networks process the "why." The functions used in these layers are dependent on the use case, but often are customizable by the user, making them significantly more robust than the average machine learning models that are often used for classification and regression, for example. The assumption in deep learning models on a fundamental level is that the data being interpreted is generated by the interactions of different factors organized in layers. As such, having multiple layers allows the model to process the data such that it builds an understanding from simple aspects to larger constructs. The objective of these models is to perform tasks without the same degree of explicit instruction that many machine learning algorithms need. With respect to how these models are used, one of the main benefits is the promise they show when applied to unsupervised learning problems, or problems where we don't know prior to performing the experiment that the response variable *y* should be given a set of explanatory variables *x*. An example would be image recognition, particularly after a model has been trained against a given set of data. Let's say we input an image of a dog in the testing phase, implying that we don't tell the model what the picture is of. The neural network will start by recognizing eyelashes prior to a snout, prior to the shape of the dog's head, and so on until it classifies the image as that of a dog.

Deep Learning Models

Now that we have established a brief overview of deep learning, it will be useful to discuss what exactly *you* will be learning in this book, as well as describe the models we will be addressing here.

This text assumes you are relatively informed by an understanding of mathematics and statistics. Be that as it may, we will briefly review all the concepts necessary to understand linear algebra, optimization, and machine learning such that we will form a solid base of knowledge necessary for grasping deep learning. Though it does help to understand all this technical information precisely, those who don't feel comfortable with more advanced mathematics need not worry. This text is written in such a way that the reader is given all the background information necessary to research it further, if desired. However, the primary goal of this text is to show readers how to apply machine learning and deep learning models, not to give a verbose academic treatise on all the theoretical concepts discussed.

After we have sufficiently reviewed all the prerequisite mathematical and machine learning concepts, we will progress into discussing machine learning models in detail. This section describes and illustrates deep learning models.

Single Layer Perceptron Model (SLP)

The *single layer perceptron* (SLP) model is the simplest form of neural network and the basis for the more advanced models that have been developed in deep learning. Typically, we use SLP in classification problems where we need to give the data observations labels (binary or multinomial) based on inputs. The values in the input layer are directly sent to the output layer after they are multiplied by weights and a bias is added to the cumulative sum. This cumulative sum is then put into an *activation* function, which is simply a function that defines the output. When that output is above or below a user-determined threshold, the final output is determined. Researchers McCulloch-Pitts Neurons described a similar model in the 1940s (see Figure 1-2).

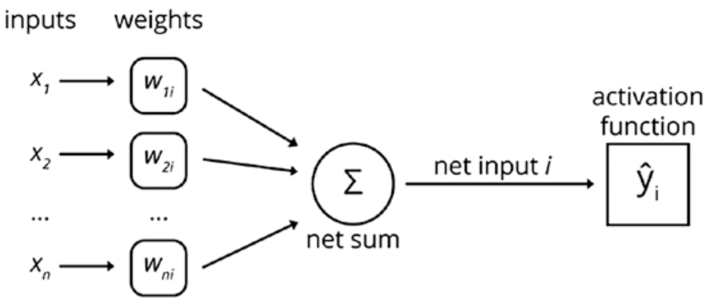

Figure 1-2. *Single layer perceptron network*

Multilayer Perceptron Model (MLP)

Very similar to SLP, the *multilayer perceptron* (MLP) model features multiple layers that are interconnected in such a way that they form a feed-forward neural network. Each neuron in one layer has directed connections to the neurons of a separate layer. One of the key distinguishing factors in this model and the single layer perceptron model is the back-propagation algorithm, a common method of training neural networks. Back-propagation passes the error calculated from the output layer to the input layer such that we can see each layer's contribution to the error and alter the network accordingly. Here, we use a gradient descent algorithm to determine the degree to which the weights should change upon each iteration. *Gradient descent*—another popular machine learning/optimization algorithm—is simply the derivative of a function such that we find a *scalar* (a number with magnitude as its only property) value that points in the direction of greatest momentum. By subtracting the gradient, this leads us to a solution that is more optimal than the one we currently are at until we reach a global optimum (see Figure 1-3).

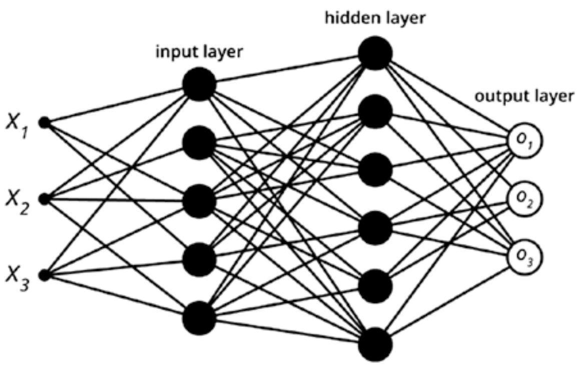

Figure 1-3. *MultiLayer perceptron network*

Convolutional Neural Networks (CNNs)

Convolutional neural networks (CNNs) are models that are most frequently used for image processing and computer vision. They are designed in such a way to mimic the structure of the animal visual cortex. Specifically, CNNs have neurons arranged in three dimensions: width, height, and depth. The neurons in a given layer are only connected to a small region of the prior layer. CNN models are most frequently used for image processing and computer vision (see Figure 1-4).

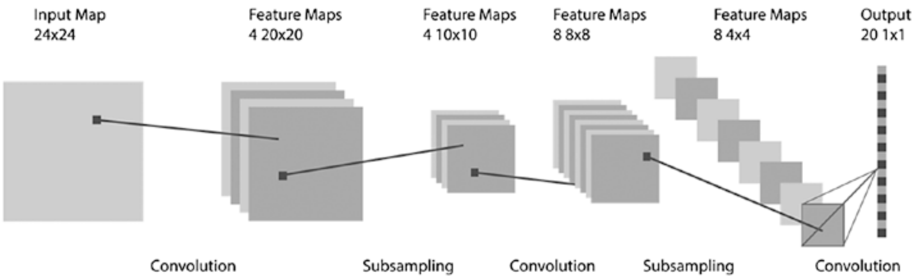

Figure 1-4. *Convolutional neural network*

Recurrent Neural Networks (RNNs)

Recurrent neural networks (RNNs) are models of *Artificial neural networks* (ANNs) where the connections between units form a directed cycle. Specifically, a *directed cycle* is a sequence where the walk along the vertices and edges is completely determined by the set of edges used and therefore has some semblance of a specific order. RNNs are often specifically used for speech and handwriting recognition (see Figure 1-5).

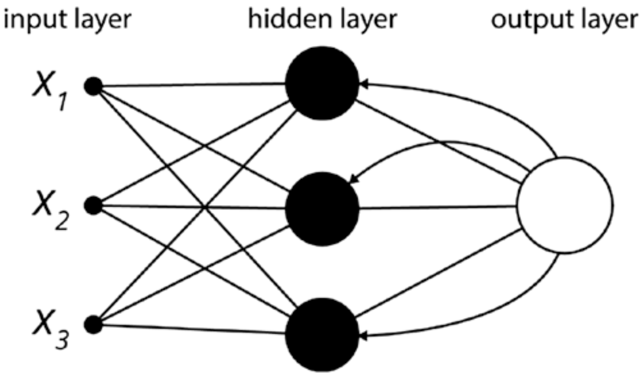

Figure 1-5. *Recurrent neural network*

Restricted Boltzmann Machines (RBMs)

Restricted Boltzmann machines are a type of binary Markov model that have a unique architecture, such that there are multiple layers of hidden random variables and a network of symmetrically coupled stochastic binary units. DBMs are comprised of a set of visible units and series of layers of hidden units. There are, however, no connections between units of the same layer. DMBs can learn complex and abstract internal representations in tasks such as object or speech recognition (see Figure 1-6).

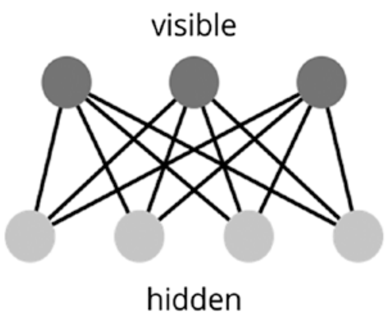

Figure 1-6. *Restricted Boltzmann machine*

Deep Belief Networks (DBNs)

Deep belief networks are similar to RBMs except each subnetwork's hidden layer is in fact the visible layer for the next subnetwork. DBNs are broadly a generative graphical model composed of multiple layers of latent variables with connections between the layers but not between the units of each individual layer (see Figure 1-7).

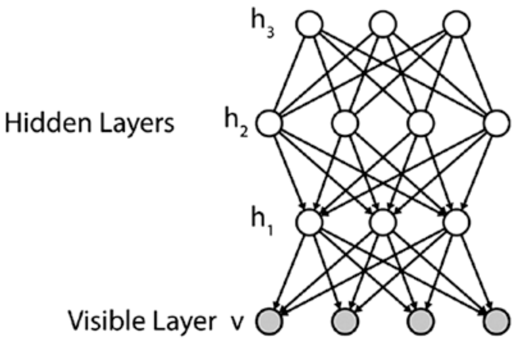

Figure 1-7. *Deep belief networks*

Other Topics Discussed

After covering all the information regarding models, we will turn to understanding the practice of data science. To aid in this effort, this section covers additional topics of interest.

Experimental Design

The emphasis of this text ultimately is to give the reader a theoretical understanding of the deep learning models such that they feel comfortable enough to apply them. As such, it is important to discuss elements of experimental design to help the reader understand proper ways to structure their research so it leads to actionable insights and not a waste of time and/or energy. Largely, I will draw upon Fisher's principles in addition to defining best practices given the problems often utilized by deep learning.

Feature Selection

A component of experimental design, but ultimately entirely a subtopic of research unto itself, I will cover the concept of variable selection and multiple methods used often by data scientists to handle high dimensional data sets. Specifically, I will speak in depth about principal components analysis as well as genetic algorithms. All the algorithms discussed are available in the R statistical language in open source packages. For those who want to research this area of research further, I'll reference papers relevant to this topic. From a deep learning perspective, we will discuss in depth how each model performs its own specific methods of feature selection by design of the layer architecture in addition to addressing recent discoveries in the field.

Applied Machine Learning and Deep Learning

For the final section of the text, I will walk the reader through using packages in the R language for machine learning and deep learning models to solve problems often seen in professional and academic settings. It is hoped that from these examples, readers will be motivated to apply machine learning and deep learning in their professional and/or academic pursuits. All the code for the examples, experiments, and research uses the R programming language and will be made available to all readers via GitHub (see the appendix for more). Among the topics discussed are regression, classification, and image recognition using deep learning models.

History of Deep Learning

Now that we have covered the general outline of the text, in addition to what the reader is expected to learn during this period, we will see how the field has evolved to this stage and get an understanding of where it seeks to go today. Although deep learning is a relatively new field, it has a rich and vibrant history filled with discovery that is still ongoing today. As for where this field finds its clearest beginnings, the discussion brings us to the 1960s.

The first working learning algorithm that is often associated with deep learning models was developed by Ivakhenenko and Lapa. They published their findings in a paper entitled "Networks Trained by the Group Method of Data Handling (GMDH)" in 1965. These were among the first deep learning systems of the feed-forward multilayer perceptron type. *Feed-forward* networks describe models where the connections between the units don't form a cycle, as they would be in a recurrent neural network. This model featured polynomial activation functions, and the layers were incrementally grown and trained by regression analysis. They were subsequently pruned with the help of a separate validation set, where regularization was used to weed out superfluous units.

In the 1980s, the neocognitron was introduced by Kunihio Fukushima. It is a multilayered artificial neural network and has primarily been used for handwritten character recognition and similar tasks that require pattern recognition. Its pattern recognition abilities gave inspiration to the convolutional neural network. Regardless, the neocognitron was inspired by a model proposed by the neurophysiologists Hubel and Wiesel. Also during this decade, Yann LeCun et al. applied the back-propagation algorithm to a deep neural network. The original purpose of this was for AT&T to recognize handwritten zip codes on mail. The advantages of this technology were significant, particularly right before the Internet and its commercialization were to occur in the late 1990s and early 2000s.

In the 1990s, the field of deep learning saw the development of a recurrent neural network that required more than 1,000 layers in an RNN unfolded in time, and the discovery that it is possible to train a network containing six fully connected layers and several hundred hidden units using what is called a wake-sleep algorithm. A *heuristic*, or an algorithm that we apply over another single or group of algorithms, a wake-sleep algorithm is a unsupervised method that allows the algorithm to adjust parameters in such a way that an optimal density estimator is outputted. The "wake" phase describes the process of the neurons firing from input to output. The connections from the inputs and outputs are modified to increase the likelihood that they replicate the correct activity in the layer below the current one. The "sleep" phase is the reverse of the wake phase, such that neurons are fired by the connections while the recognitions are modified.

As rapidly as the advancements in this field came during the early 2000s and the 2010s, the current period moving forward is being described as the watershed moment for deep learning. It is now that we are seeing the application of deep learning to a multitude of industries and fields as well as the very devoted improvement of the hardware used for these models. In the future, it is expected that the advances covered in deep learning will help to allow technology to make actions in contexts where humans often do today and where traditional machine learning algorithms have performed miserably. Although there is certainly still progress to be made, the investment made by many firms and universities to accelerate the progress is noticeable and making a significant impact on the world.

Summary

It is important for the reader to ultimately understand that no matter how sophisticated any model is that we describe here, and whatever interesting and powerful uses it may provide, there is no substitute for adequate domain knowledge in the field in which these models are being used. It is easy to fall into the trap, for both advanced and introductory practitioners, of having full faith in the outputs of the deep learning models without heavily evaluating the context in which they are used. Although seemingly self-evident, it is important to underscore the importance of carefully examining results and, more importantly, making actionable inferences where the risk of being incorrect is most limited. I hope to impress upon the reader not only the knowledge of where they can apply these models, but the reasonable limitations of the technology and research as it exists today.

This is particularly important in machine learning and deep learning because although many of these models are powerful and reach proper solutions that would be nearly impossible to do by hand, we have not determined *why* this is the case always. For example, we understand how the back-propagation algorithm works, but we can't see it operating and we don't have an understanding of what exactly happened to reach such a conclusion. The main problem that arises from this situation is that when a process breaks, we don't necessarily always have an idea as to why. Although there have been methods created to try and track the neurons and the order in which they are activated, the decision-making process for a neural network isn't always consistent, particularly across differing problems. It is my hope that the reader keeps this in mind when moving forward and evaluates this concern appropriately when necessary.