EAI/Springer Innovations in Communication and Computing

Kolla Bhanu Prakash
Ramani Kannan
S. Albert Alexander
G. R. Kanagachidambaresan *Editors*

Advanced Deep Learning for Engineers and Scientists

A Practical Approach

EAI
RESEARCH MEETS INNOVATION

Springer

EAI/Springer Innovations in Communication and Computing

Series Editor

Imrich Chlamtac, European Alliance for Innovation, Ghent, Belgium

Editor's Note

The impact of information technologies is creating a new world yet not fully understood. The extent and speed of economic, life style and social changes already perceived in everyday life is hard to estimate without understanding the technological driving forces behind it. This series presents contributed volumes featuring the latest research and development in the various information engineering technologies that play a key role in this process.

The range of topics, focusing primarily on communications and computing engineering include, but are not limited to, wireless networks; mobile communication; design and learning; gaming; interaction; e-health and pervasive healthcare; energy management; smart grids; internet of things; cognitive radio networks; computation; cloud computing; ubiquitous connectivity, and in mode general smart living, smart cities, Internet of Things and more. The series publishes a combination of expanded papers selected from hosted and sponsored European Alliance for Innovation (EAI) conferences that present cutting edge, global research as well as provide new perspectives on traditional related engineering fields. This content, complemented with open calls for contribution of book titles and individual chapters, together maintain Springer's and EAI's high standards of academic excellence. The audience for the books consists of researchers, industry professionals, advanced level students as well as practitioners in related fields of activity include information and communication specialists, security experts, economists, urban planners, doctors, and in general representatives in all those walks of life affected ad contributing to the information revolution.

Indexing: This series is indexed in Scopus, Ei Compendex, and zbMATH.

About EAI

EAI is a grassroots member organization initiated through cooperation between businesses, public, private and government organizations to address the global challenges of Europe's future competitiveness and link the European Research community with its counterparts around the globe. EAI reaches out to hundreds of thousands of individual subscribers on all continents and collaborates with an institutional member base including Fortune 500 companies, government organizations, and educational institutions, provide a free research and innovation platform.

Through its open free membership model EAI promotes a new research and innovation culture based on collaboration, connectivity and recognition of excellence by community.

More information about this series at http://www.springer.com/series/15427

Kolla Bhanu Prakash • Ramani Kannan
S. Albert Alexander
G. R. Kanagachidambaresan
Editors

Advanced Deep Learning for Engineers and Scientists

A Practical Approach

Springer

EAI
RESEARCH MEETS INNOVATION

Editors
Kolla Bhanu Prakash
KL Deemed to be University
Vijayawada, AP, India

S. Albert Alexander
Kongu Engineering College
Perundurai, Tamil Nadu, India

Ramani Kannan
Electrical & Electronics Engineering
Universiti Teknologi Petronas
Seri Iskandar, Perak, Malaysia

G. R. Kanagachidambaresan
Department of CSE
Vel Tech Rangarajan Dr Sagunthala R&D
Institute of Science and Technology
Chennai, Tamil Nadu, India

ISSN 2522-8595 ISSN 2522-8609 (electronic)
EAI/Springer Innovations in Communication and Computing
ISBN 978-3-030-66518-0 ISBN 978-3-030-66519-7 (eBook)
https://doi.org/10.1007/978-3-030-66519-7

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

To our family, students, scholars, and dear friends

Preface

Deep learning is the latest buzzword to understand and utilize new sensor technology and is a primary tool for smart city development. New adaptation and infrastructure development in developing countries are mainly influenced by computer technologies such as deep learning and machine learning approaches. Deep learning methods can cover wider engineering disciplines with various real-world applications such as speech recognition, computer vision, and smart computing scenarios. These methods require broad domain knowledge of various subjects; for example, advanced driver-assistance system (ADAS) design requires the knowledge of maths, physics, computer vision (CV), mechanics, and sensor technology. Various deep learning concepts about healthcare, smart systems, recommendation systems, etc., with real-time discussions are covered in this book. Deep learning is capable of working with unstructured data that requires more time for humans to extract the relevant information. Industry 4.0 standards have perceived the power of deep learning methodology and are utilizing the artificial intelligence (AI) systems for cyber-physical system (CPS). Deep learning is a subset of machine learning in artificial intelligence (AI) that utilizes an artificial neural network (ANN) to operate CPS. The current pandemic situation has attracted several researchers to study the design of healthcare-oriented systems using machine learning algorithms. The outbreak of vector-borne diseases – such as chikungunya, malaria, and dengue – predicted using machine learning as well as its spread in the Indian subcontinent is elucidated in detail. During this covid-19 period, researchers focus on ways for the identification of increase in cholesterol level as well as for contactless sensing and diagnosis. The prediction of eukaryotic plasma cholesterol from human G protein-coupled receptors (GPCRs) is studied through K-nearest neighbor (KNN) and support vector machine (SVM) approach. In many cases, a hybrid approach is required to balance the drawbacks of each approach and to derive a potential application for better prediction. In this book, the hybrid KNN–SVM approach is shown to examine the pattern of cholesterol level, and the obtained data is validated with experimental studies. The detection of early-stage diabetic retinopathy using convolutional neural network (CNN) approach is also addressed. Due to its novelty and accuracy, the retinal scan comparison is performed with the data set to assess the present eye condition of a person. The most widely used deep learning techniques, such as CNN, recurrent neural networks (RNN), Tensorflow, Python tool, deep convolutional generative adversarial network (DCGAN), Auto-encoders, long short-term memory (LSTM), and gated recurrent unit (GRU), are discussed. The identification of image and text using these algorithms, their working perspectives, and their contextual speed are explained for readers. Deep learning in bioinformatics enables to identify hidden information and make correct predictions. Omics data analysis, protein structure prediction, and biomedical image processing are explained with real-time data set. Character recognition and Opencv are mostly used in CPS for automation and machine-to-machine interaction. Both Opencv and character recognition algorithms are capable of operating in single-board computers (SBCs) and have a wide application in the current Industry 4.0 era. CNN-based document analysis, document recognition, scene-text classification, and localizations were discussed with real-time results, and the comparison is also done with standard benchmark algorithms. This book provides an overview of deep learning and machine learning techniques for better prediction and smart computing environments.

Vijayawada, AP, India — Kolla Bhanu Prakash
Seri Iskandar, Perak, Malaysia — Ramani Kannan
Perundurai, Tamil Nadu, India — S. Albert Alexander
Chennai, Tamil Nadu, India — G. R. Kanagachidambaresan

Acknowledgments

We thank the Almighty, our parents, and our spouses for the endless support, guidance, and love through all our life stages. We are thankful to our beloved family members for standing beside us throughout our career, to move our career forward through editing this book.

We specially thank Sri. Koneru Satyanarayana, President, KL University, India, for his continuous support and encouragement for the preparation of this book. We dedicate this book to him, our collaborators, and our family members.

Our great thanks to our students, who have put in their time and effort to support and contribute in some manner. We express our gratitude toward all who supported, shared, talked things over, read, wrote, offered comments, allowed us to quote their remarks, and assisted in editing, proofreading, and designing through this book journey.

We believe that the team of authors provides the perfect blend of knowledge and skills that went into authoring this book. We thank each of the authors for devoting their time, patience, perseverance, and effort toward this book; we think that it will be a great asset to all researchers in this field!

We are grateful to the Springer Publishing team, who showed us the ropes to write this book. Without their support and knowledge, we would not have ventured into starting and creating this book. Their trust in us and their guidance – providing the necessary time and resources – gave us the freedom to manage this book.

Last but not least, we thank our readers, who gave us their trust – we hope our work inspires and guides them.

Kolla Bhanu Prakash
Ramani Kannan
S. Albert Alexander
G. R. Kanagachidambaresan

About the Book

This book contains 11 important contributions addressing current real-time problems. This pandemic period has attracted several researchers on building many healthcare-related projects and algorithms on disease prediction and diagnosis approaches. Deep learning is able to work with unstructured input which requires more time for humans to extract the relevant information. Industry 4.0 CPS standards have perceived the power of deep learning methodology and are utilizing the AI systems for industrial machines and smart city development. This book will be a useful resource to recognize and understand the potential of deep learning for researchers and engineers working on smart city projects and healthcare domain.

Contents

About the Editors

Kolla Bhanu Prakash is working as a professor and research group head in CSE Department, KL University, Vijayawada, Andhra Pradesh, India. He received his MSc and MPhil in Physics from Acharya Nagarjuna University, Guntur, India, and ME and PhD in Computer Science Engineering from Sathyabama University, Chennai, India. He has more than 15 years of experience working in academia, research, teaching, and academic administration. His research interests include artificial intelligence, deep learning, data science, smart grids, cyberphysical systems, cryptocurrency, blockchain technology, and image processing. He is IEEE Senior Member and Fellow-ISRD, Treasurer – ACM Amaravathi Chapter, India, LMISTE, MIAENG, and SMIRED. He has reviewed more than 125 peer-reviewed journals, which are indexed in Publons. He is the editor for six books in Elsevier, CRC Press, Springer, Wiley, and De Gruyter publishers. He has published 65 research papers, 5 patents, 5 books, and 4 accepted. His Scopus H-index is 12. He is a frequent editorial board member and TPC member in flagship conferences and refereed journals. He is a reviewer for Inderscience Publishers and journals such as *IEEE Access*, *Springer Nature*, *Applied Soft Computing Journal* (Elsevier), *Wireless Networks*, *IET*, *KSII*, and *IEEE Computer Society*. He is a series editor of *Next Generation Computing & Communication Engineering* (Wiley); under this series, he had signed agreements for five books.

Ramani Kannan is a senior lecturer in Universiti Teknologi PETRONAS, Malaysia. He received his BE from Bharathiyar University, India. Later, he completed his ME and PhD in Power Electronics and Drives from Anna University. He holds more than 120 publications in reputed international and national journals and conferences. He is a chartered engineer (CEng, UK), active senior member in IEEE (USA), and members of IE(I), IET(UK), ISTE(I), and Institute of Advanced Engineering and Science (IAENS). He has received several awards such as Career Award for Young Teacher from AICTE, India, in 2012; Young Scientist Award in power electronics and Drives, 2015; Highest Research publication Award in 2017; Award for Outstanding Performance, Service and Dedication in 2019 at UTP, Malaysia; Outstanding Researcher Award in UTP Q Day 2019; Best Presenter

Award, IEEE-CENCON 2019 international conference in Indonesia. He is actively serving as a secretary for IEEE Power Electronics Society, Malaysia, since 2020.

He is the editor-in-chief of *Asian Scientific Research* (2011–2018) and regional editor of *International Journal of Computer Aided Engineering and Technology*, Inderscience Publisher, UK, since 2015. He is an associate editor for journals *IEEE Access* since 2018 and *Advanced Materials Science and Technology since 2020.* He is serving as a guest editor for Elsevier, Inderscience, IGI Global, CRC, Taylor & Francis, Bentham Science, and IJPAM. His research interests involve power electronics, inverters, modeling of induction motor, artificial intelligence, machine learning, and optimization techniques.

S. Albert Alexander is a postdoctoral research fellow from Northeastern University, Boston, MA, USA. He is the recipient of prestigious Raman Research Fellowship from the University Grants Commission (Government of India). His research focuses on fault diagnostic systems for solar energy conversion systems and smart grids. He has 14 years of academic and research experience. He has published 25 technical papers in international and national journals (including IEEE Transactions, IET, Elsevier, Taylor & Francis, and Wiley) and presented 23 papers at national and international conferences.

He has completed four projects funded by the Government of India, and one multilateral project is in progress with the overall grant amount of Rs.103 lakhs. His PhD work on power quality earned him a National Award from ISTE, and he has received 25 awards for his meritorious academic and research career (such as Young Engineers Award from IE(I) and Young Scientist Award from SPRERI, Gujarat). He has also received the National Teaching Innovator Award from MHRD (Government of India). He is an approved "Margadarshak" from AICTE (Government of India). He is the approved Mentor for Change under Atal Innovation Mission.

He has guided 32 graduate and postgraduate projects. He is presently guiding seven research scholars and one completed his PhD under his guidance. He is a member and holds prestigious positions in various national and international forums (such as a senior member of IEEE and vice president of Energy Conservation Society, India). He has been an invited speaker in 215 programs both in India and the USA. He has organized 12 events, including faculty development programs, workshops, and seminars.

He has completed his graduate program in Electrical and Electronics Engineering from Bharathiar University and his postgraduate program from Anna University, India. Presently, he is working as an associate professor in the Department of EEE, Kongu Engineering College, and also doing research work in artificial intelligent controllers, smart healthcare systems, smart grids, solar PV, and power quality improvement techniques. He has authored the following books in his areas of interest: (1) *Power Electronic Converters for Solar PV Systems*, Elsevier; (2) *Computational Paradigm Techniques for Enhancing Electric Power Quality*, CRC Press; (3) *Basic Electrical, Electronics and Measurement Engineering*, Anuradha Publishers; and (4) *Special Electrical Machines*, Anuradha Publishers.

G. R. Kanagachidambaresan received his BE in Electrical and Electronics Engineering from Anna University in 2010 and ME in Pervasive Computing Technologies from Anna University in 2012. He has completed his PhD (Wireless Body Area Networks and Healthcare) from Anna University, Chennai, in 2017. He is currently an associate professor in the Department of CSE in Veltech Rangarajan Dr Sagunthala R&D Institute of Science and Technology. His research areas of interest include body sensor network and fault-tolerant wireless sensor network. He has published several articles in reputed journals and has undertaken several consultancy activities for leading MNC companies. He is serving as editorial review board members for peer-reviewed journals and has also guest-edited several special issue volumes and books in Springer. He is presently working on several government-sponsored research projects of ISRO, DBT, and DST.

Introduction to Deep Learning

R. Indrakumari, T. Poongodi, and Kiran Singh

1 Introduction

The human brain is the incredible organ that dictates the signals received from sound, sight, smell, touch, and taste. The brain stores emotions, experiences, memories, and even dreams. The brain takes decisions and solves many problems that even the powerful supercomputers lack [1]. Based on this, researchers are dreamed of constructing intelligent machines like the brain. Later researchers invented robots to assist human activities, automatic disease detection microscopes, and self-driving cars. These inventions still required human interventions to do some computational problems. To tackle this problem, researchers want to build a machine that can learn by themselves and solve more complex problems in the speed of the human brain. These necessities pave the way to the most active field of artificial machine intelligence called deep learning.

2 Neurons

The basic unit of the human brain is the neurons. Very small portions of the brain, about the size of wheat, have over 10,000 neurons with more than 6000 connections with other neurons [2]. The information perceived by the brain is captured by the neurons, and the same is passed from a neuron to others for processing, and the final result is sent to other cells. It is depicted in Fig. 1. Dendrites are an antenna-like structure in the neurons that receives the inputs. Based on the frequency of usage,

R. Indrakumari (✉) · T. Poongodi · K. Singh
School of Computing Science and Engineering, Galgotias University,
Greater Noida, Uttar Pradesh, India

K. B. Prakash et al. (eds.), *Advanced Deep Learning for Engineers and Scientists*, EAI/Springer Innovations in Communication and Computing,
https://doi.org/10.1007/978-3-030-66519-7_1

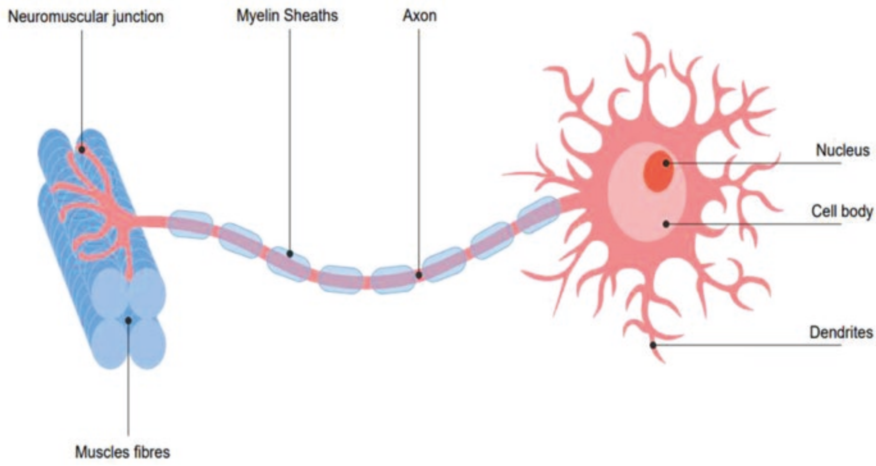

Fig. 1 Biological neuron's structure

the inputs are classified into strengthened and weakened. The connection strength estimates the involvement of the input pertaining to the neuron's output. The input signals are weighted by the connection strength and summed collectively in the cell body. The calculated sum takes the form of a new signal, and it is thriven along the cell's axon to reach the destination neurons.

In 1943, Warren S. McCulloch and Walter H. Pitts [3] concentrated on the functional understanding of the neurons that exist in the human brain and created a computer-based artificial model as shown in Fig. 2.

As in the biological neurons, the artificial neuron receives inputs x_1, x_2, $x_3 \ldots . x_n$, and respectively input is multiplied by particular weights w_1, w_2, $w_3, \ldots . w_n$, and the calculated sum is considered to make the *logit* of the neuron:

$$Z = \sum_{i=0}^{n} w_i x_i \tag{1}$$

Some logit may include a constant value called the bias. Finally, the logit is passed through a function f to make the desired output y = f (z).

3 History of Deep Learning

The history of deep learning started in the early 1940s when Warren McCulloch and Walter Pitts developed a computer model focusing on the human neural system. They applied mathematics and algorithms and called it "threshold logic" to imitate the thinking process. Deep learning is a subsequent derivative of machine learning that applies algorithms, processes the data, and develops abstractions. Various

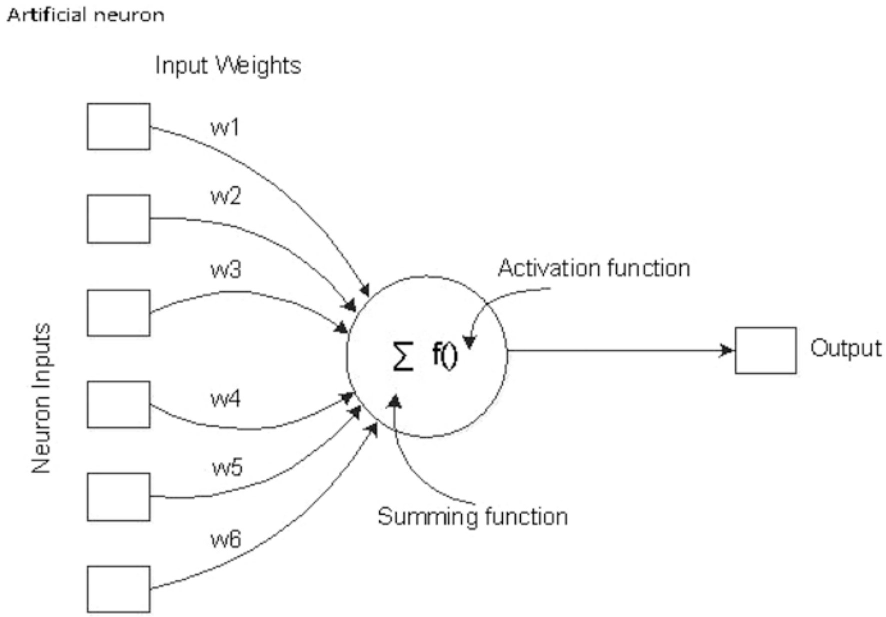

Fig. 2 Neuron in an artificial neural net

algorithms are applied to process data, to recognize objects and human speech. The output of the former layer is provided as the input to the next layer.

In 1960 Henry J. Kelley has started to develop the Backpropagation Model and was extended by Stuart Dreyfus in 1962. The early version of Backpropagation was not so efficient and clumsy. Following this, in 1965, Valentin Grigor'evich Lapa has proposed cybernetics and forecasting techniques, and Alexey Grigoryevich Ivakhnenko has proposed the data handling methodology using polynomial activation functions. The best feature chosen statistically is forwarded to the next layer manually.

Kunihiko Fukushima has developed the first convolutional neural networks with multiple pooling and convolutional layers. Later in 1979, he developed neocognitron, a multilayered and hierarchical artificial neural network design that can recognize visual patterns. Neocognitron is said to be the best model at that time as it uses new learning methods with top-down connections. It contains the selective attention model which recognizes the individual patterns. The developed neocognitron can be able to identify the unknown and missing information with a concept called inference.

In the late 1970s, Seppo Linnainmaa wrote a Fortran code for backpropagation. In 1985, Williams and Hinton studied that backpropagation can provide "interesting" distribution representations. Yann LeCun combined backpropagation with convolutional neural networks and showed the first practical demonstration to read "handwritten" digits at Bell Labs in 1989. Later many optimistic researchers

Fig. 3 Roadmap of deep learning history

exaggerated artificial intelligence; notably in 1995, Dana Cortes and Vladimir Vapnik have proposed a model to map and identify similar data, called the support vector machine. In 1997, Sepp Hochreiter and Juergen Schmidhuber have proposed long short-term memory (LSTM) for recurrent neural networks (Fig. 3).

The new era for deep learning began in 1999 as it is the evolution of graphics processing units (GPUs). In 2000, the vanishing gradient problem is identified which paved the way for the development of long short-term memory. Fei-Fei Li an AI expert assembled ImageNet which can process more than 14 million labeled images. During 2011 and 2012, AlexNet a convolutional neural network won many international competitions. In 2012, Google Brain announced a project called The Cat Experiment, which overcomes the limitations of unsupervised learning. At present, the evolution of artificial intelligence and the processing of big data are dependent on deep learning.

4 Feed-Forward Neural Networks

The neurons in the human brain are arranged as layered structure, and even most of the human intelligence part in the brain, the cerebral cortex, is of six layers [4]. The perceived information travels from layer to another layer until obtaining the conceptual understanding from the sensory input.

In Fig. 4, a three-layer perceptron is shown with the hidden layer that contains neurons with nonlinear activation functions. Arbitrarily complex decision and computation of any likelihood function can be easily done by a three-layer perceptron.

From Fig. 4, it is noted that the connection traverses from the low-level layer to the high-level layer and there are no communications among neurons which exist in the same layer as well from the higher to the lower level. Hence these setup is called the feed-forward networks. The middle layer in Fig. 4 is the hidden layer where the magic happens when the neural network tries to solve complex problems. Every layer in Fig. 4 has an equal number of neurons, which is not mandatory. The input and output are represented as vectors. Linear neurons are represented by a linear function in the form of $f_z = a_z + b$. Linear neurons are easy to compute but restricted with limitations. A feed-forward network with only linear neurons contains no hidden layer which enables the users to get vital features from the input layer. In practice, there are three possible types of neurons, namely, sigmoid neuron, tanh neurons, and ReLU neurons, that dumped the nonlinearity concept. The sigmoid neurons use the function

$$f = \frac{1}{1 + e^{-z}} \tag{2}$$

The above equation represents that when the value of logit is actually small, then the output is very close to 0, and it is 1 when the value of logistic is very large. Between the values 0 and 1, the neuron takes the shape of S as shown in Fig. 4.

Based on the types of connections the neural network architecture is categorized into “recurrent neural networks” in which there exists a synaptic connection from output to the input whereas in “feed-forward neural networks” there exists a feedback operation from output to inputs. Neural networks are constructed as either single layers or multilayer.

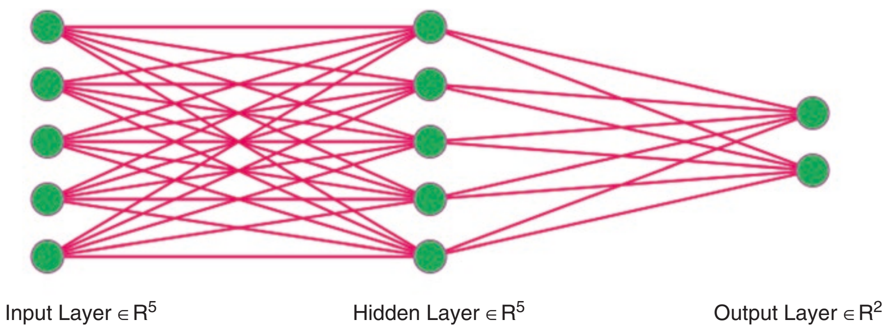

Fig. 4 Three-layer perceptron network with continuous inputs, two output, and two hidden layers

4.1 Backpropagation

Backpropagation is the heart of neural network training which fine-tunes the weights of neural net obtained in the previous epoch. It was developed in 1970, and researchers fully appreciated it after 1986 when David Rumelhart, Geoffrey Hinton, and Ronald Williams published a paper describing that backpropagation works faster and provides solutions for previously unsolved problems. Backpropagation is a kind of supervised learning method for multilayer artificial neural networks (ANNs) with applications ranging from classification, pattern recognition, medical diagnostics, etc. The backpropagation algorithm made the multilayer perceptron networks occupy a place in the neural network's research toolbox. The multilayer perceptron is perceived as a feed-forward network with more than one layer of nodes between the input and output nodes. It updates the synaptic weights by propagating a gradient vector back to the input in which the elements are defined as the derivative of an error measure for a parameter. The error signals are the significant difference between the actual and the desired outputs.

The backpropagation algorithms are considered as a generalized view of the least-mean-square (LMS) algorithm that consists of a forward pass and a backward pass. The backpropagation computes specifically all the partial derivatives $\frac{\partial f}{\partial w_i}$ where w_i is the ith parameter and f is the output.

Consider a multilayer feed-forward neural network as shown in Fig. 2. Let us assume a neuron i is present in the output layer and the error signal for n^{th} iteration is given by the equation

$$ei(m) = di - yi(m) \tag{3}$$

where d_i is the desired output for neuron i and $y\,j\,(m)$ is the actual output for neuron i, computed using the current weights of the network at iteration m.

Equation 2 represents the instant error energy value y for the neuron i as

$$\varepsilon_i(m) = \frac{1}{2} e_i^2(m) \tag{4}$$

The instantaneous value $\varepsilon_i(m)$ is the sum of all $\varepsilon_i(m)$ for all neurons in the output layer as represented in Eq. 3

$$\varepsilon_i(m) = \frac{1}{2} \sum_{i \varepsilon S} e_i^2(m) \tag{5}$$

where S is the set of all neurons present in the output layer. For consideration, suppose a training set contains N patterns, and the average square energy for the network is given by Eq. 4:

$$\varepsilon_{avg} = \frac{1}{N}\sum_{n=1}^{N}\varepsilon(m) \tag{6}$$

The modes of backpropagation algorithms are (a) batch mode and (b) sequential mode. In the batch mode, the weight updates are done after an epoch is completed. In contrast to this, the sequential mode or stochastic mode updates are performed after the presentation of each training example. The following equation gives the output expression for the neuron i

$$y_i(m) = f\left[\sum_{i=0}^{n} w_{ij}(m)\, y_i(m)\right] \tag{7}$$

where n represents the total number of inputs to the neuron i from the previous layer and f is the activation function used in the neuron i.

The updated weight to be applied to the weights of the neuron i is directly proportional to the partial derivative of the instantaneous error energy $\varepsilon(n)$ for the corresponding weight, and it is represented as

$$\frac{\partial \varepsilon(m)}{\partial w_{ij}(m)} \tag{8}$$

Using the chain rule of calculus, it is expressed as

$$\frac{\partial \varepsilon(m)}{\partial w_{ij}(m)} = \frac{\partial \varepsilon(m)}{\partial e_i(m)}\frac{\partial e_i(m)}{\partial y_i(m)}\frac{\partial y_i(m)}{\partial w_{ij}(m)} \tag{9}$$

Equation 10 is obtained from Eqs. (2), (1), and (5)

$$\frac{\partial \varepsilon(m)}{\partial e_i(m)} = e_i(m) \tag{10}$$

$$\frac{\partial e_i(m)}{\partial y_i(m)} = -1 \tag{11}$$

$$\begin{aligned}\frac{\partial y_i(m)}{\partial w_{ij}(m)} &= f'\left[\sum_{i=0}^{m} w_{ij}(m)\, y_i(m)\right]\frac{\partial\left[\sum_{i=0}^{m} w_{ij}(m)\, y_i(m)\right]}{\partial w_{ij}(m)}\\ &= f'\left[\sum_{i=0}^{m} w_{ij}(m)\, y_i(m)\right] y_i(m)\end{aligned} \tag{12}$$

where

$$f'\left[\sum_{i=0}^{m} w_{ij}(m)\, y_i(m)\right] = \frac{\partial f\left[\left[\sum_{i=0}^{m} w_{ij}(m)\, y_i(m)\right]\right]}{\partial\left[\left[\sum_{i=0}^{m} w_{ij}(m)\, y_i(m)\right]\right]}$$

Substituting Eqs. (8), (9), and (10) in Eq. 9, the following expression arrives

$$\frac{\partial \varepsilon(n)}{\partial w_{ij}(m)} = -e_j(m)\, f'\left[\sum_{i=0}^{m} w_{ij}(m)\, y_i(m)\right] y_i(m) \tag{13}$$

Delta rule is used to provide the correction $\Delta w_{ij}(m)$, and it is expressed as

$$\Delta w_{ij}(m) = -\eta \frac{\partial \varepsilon(n)}{\partial w_{ij}(m)} \tag{14}$$

where η is a constant pre-determined parameter for the learning rate in the back-propagation algorithm.

5 Types of Deep Learning Networks

The deep learning network is classified into three classes depending upon the techniques and architectures used for a particular application like synthesis, classification, and recognition. They are classified into:

(i) Unsupervised deep learning network
(ii) Supervised deep learning network
(iii) Hybrid deep learning networks

Unsupervised deep learning network captures higher-order correlation data for synthesis purposes when there is no clear target class defined. In supervised learning of deep networks, discriminative power is provided for pattern classification by portraying the distributions of classes accustomed on the data which is visible. It is otherwise known as discriminative deep networks. A hybrid deep neural network exploits both discriminative and generative components. Moreover, a hybrid deep neural network model is structured by converging homogeneous convolution neural network (CNN) classifiers. The CNN classifiers are trained to yield an output as one for the predicted class and zero for all the other classes.

6 Deep Learning Architecture

In this deep learning architecture section, the commonly used deep learning approaches are discussed. Representation is a significant factor in deep learning. In the traditional method, the input features are extracted from raw data to be fed in machine learning algorithms. It relies on domain knowledge and the practitioner's expertise to determine the pattern. Traditional Software Engineering methodology like create, analyze, select, and evaluate are time-consuming and laborious. In contrast, the appropriate features are learned from the data directly without any human intervention and facilitate the discovery of dormant relationship among data that might be otherwise hidden or unknown.

In deep learning, the complex data representation is commonly expressed as compositions of simpler representations. Most of the deep learning algorithms are constructed based on the conceptual framework of artificial neural network (ANN), and it comprises interconnected nodes called as "neurons" which are organized in layers shown in Fig. 5. The neuron which does not exist in these two layers is called hidden units, and it stores the set of weights W.

Artificial neural network weights can be augmented by minimizing the loss function, for instance, negative log-likelihood, and it is denoted in Eq. 1:

$$E(\theta,D) = -\sum_{i=0}^{D}\left[\log P\left(Y = y_i|,x_i|,\theta\right)\right] + \lambda \quad \theta \quad p \tag{15}$$

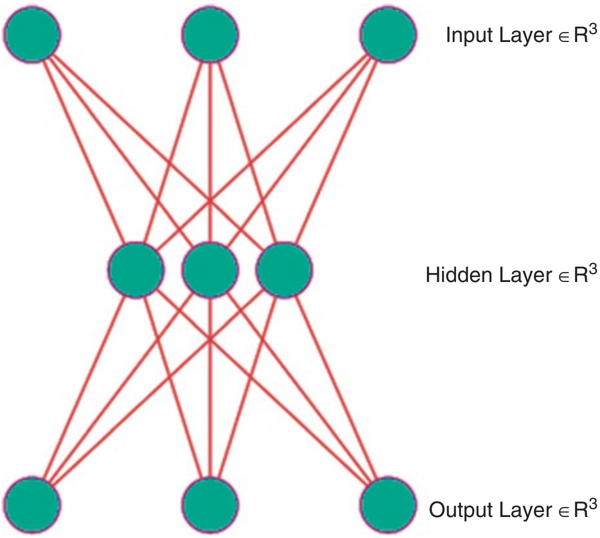

Fig. 5 Neural network with 1, 2, 1 input, hidden, and output layers

The first term minimizes the total log loss in the whole training dataset D.

The second term minimizes the p-norm of learned parameter θ_i, and it is controlled by λ a tunable parameter.

It is referred as regularization, and it prevents a model to be overfitting. Normally, the loss function can be optimized using a backpropagation mechanism, and it is meant for weight optimization that reduces the loss by traversing backward from the final layer in the network. Some of the deep learning open-source tools are Keras3, Theano2, TensorFlow1, Caffe6, DeepLearning4j8, CNTK7, PyTorch5, and Torch4. Some commonly used deep learning models discussed are based on optimization strategy and ANN's architecture. The deep learning algorithms are categorized into supervised and unsupervised techniques. The supervised deep learning architecture includes convolutional neural networks, multilayer perceptrons, and recurrent neural networks. The unsupervised deep learning architecture includes autoencoders and restricted Boltzmann machines (Fig. 6).

6.1 Supervised Learning

6.1.1 Multilayer Perceptron (MLP)

Multilayer perceptron holds many hidden layers; the neurons in the base layer *i* is completely connected to neurons in *i + 1* layer. Such type of network is restricted to have minimal hidden layers, and the data is allowed to transmit in one direction only. A weighted sum is computed for the outputs received from the hidden layer in each hidden unit. Equation 16 represents a nonlinear activation function σ of the computed sum. At this point, d refers to the number of units available in the previous layer, and x_j is referred as the output received from the previous layer j_{th} node. b_{ij} and w_{ij} are considered as bias and weight terms that are associated with each x_{ij}. Tanh or

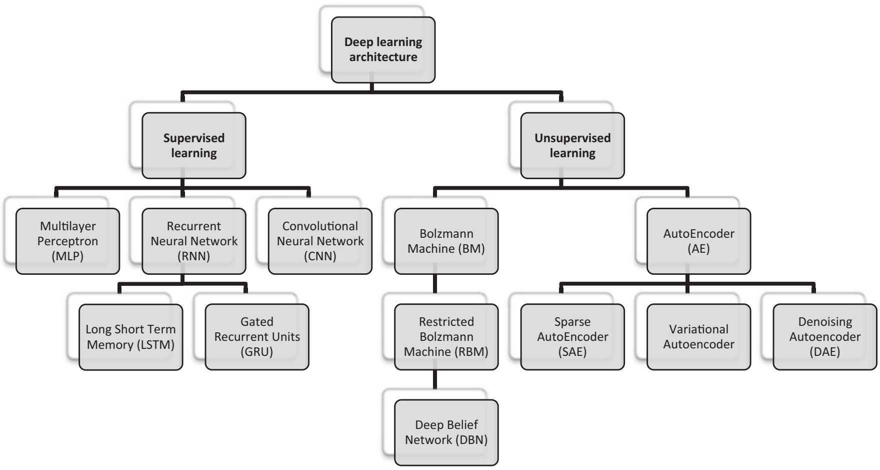

Fig. 6 Deep learning architecture and output layers

sigmoid are taken as the nonlinear activation functions in the conventional network, and rectified linear units (ReLU) [8] are used in modern networks.

A multilayer perceptron comprises of multiple hidden layers where

$$hi = \sigma\left(\sum_{j=1}^{d} x_j w_{ij} + b_{ij}\right) \tag{16}$$

After optimizing hidden layer weights during training, a correlation among the input x and output y is learned. The availability of many hidden layers makes the input data representation in a high-level abstract view because of the hidden layer's nonlinear activations. It is one of the simplest models among other learning architectures which incorporate completely connected neurons in the final layer.

6.1.2 Recurrent Neural Network (RNN)

CNN is an appropriate choice if the input data has a neat spatial structure (e.g., collection of pixels in an image), and RNN is a logical choice if the input data is ordered sequentially (e.g., natural language or time series data). One-dimensional sequence is fed into a CNN; the output of the extracted features will be shallow [8], meaning only closed localized relationships among few neighbors are considered for feature representations. RNNs are capable of handling long-range temporal dependencies. In RNN, hidden state *ht* is updated based on the triggering of current input *xt* at a time *t* and the previously hidden state *ht-1*. Consequently, the final hidden state contains complete information from all of its elements after processing an entire sequence. RNN includes:

1. Long short-term memory (LSTM)
2. Gated recurrent units (GRU)

The symbolic representation of RNN is shown in Fig. 7, with its equivalent extended representation, for instance, three input units, three hidden units, and an output. The input time step is united with the present hidden state that depends on the previous hidden state.

RNN includes LSTM and GRU models, the most popular variants referred to as gated RNN. The conventional RNN consists of interconnected hidden units, whereas

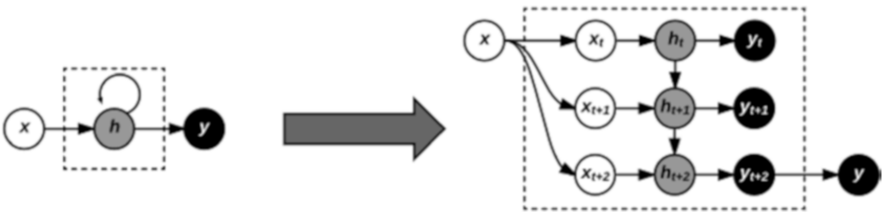

Fig. 7 RNN with extended representation

a gated RNN is substituted by a cell that holds an internal recurrence loop, and significantly the gates in this model control the information flow. The main advantage of gated RNN lies in modelling longer-term sequential dependencies.

6.1.3 Convolutional Neural Network (CNN)

CNN is a famous tool in recent years, particularly in image processing, and are stirred by the organization of the cat's visual cortex [5]. The local connectivity is imposed on the raw data on CNN. For example, more significant features are extracted by perceiving the image as a group of local pixel patches rather considering 50 x 50 image as individual 2500 unrelated pixels. A one-dimensional time series may also be viewed as a set of local signal segments. In particular, the equation for one-dimensional convolution is given as

$$C_{1d} = \sum_{a=-\infty}^{\infty} x(a).w(t-a) \tag{17}$$

where x refers to the input signal and w refers to the weight function or convolution filter.

The equation for two-dimensional convolution is given, where k is a kernel and X is a 2D grid:

$$C_{2d} = \sum_{m}\sum_{n} X(m,n)K(i-m,j-n) \tag{18}$$

The feature maps are extracted by calculating the weights of the input in a filter or a kernel. CNN encompasses sparse interactions considered as filters normally smaller than the input that results in less number of parameters. Parameter sharing is correspondingly encouraged in CNN because every filter is functional to the entire input. However, in CNN the same input is received from the previous layer which perfectly learns several lower level features. Subsampling is applied to aggregate the features which are extracted. The CNN architecture consists of two convolutional layers trailed by a pooling layer as depicted in Fig. 8. The application of CNNs is best in computer vision [6, 7].

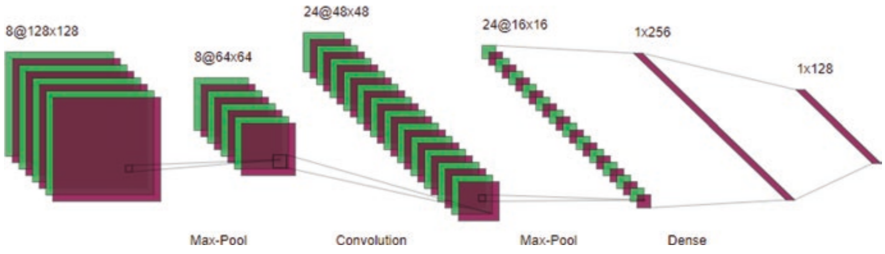

Fig. 8 ConvNet and output layers

6.2 Unsupervised Learning

6.2.1 Autoencoder (AE)

Autoencoder (AE) is the deep learning model that exemplifies the concept of unsupervised representation learning. Initially, it has pertained to supervised learning models once the labeled data was limited, but it is still remained to be useful for complete unsupervised learning such as phenotype discovery. In AE, the input is encoded into a lower-dimensional space z, and it is decoded further by reconstructing $\overline{x}$ of the corresponding input x. Hence, the encoding and decoding processes of an encoder are respectively given in equation with a single hidden layer. The encoding and decoding weights are represented as W and W0, and the reconstruction error is minimized. Z is a reliable encoded representation.

$$\mathrm{z} = \sigma\left(Wx + b\right) \tag{19}$$

$$\overline{x} = \sigma\left(W'z + b'\right) \tag{20}$$

As soon as an AE is well trained, then a single input is fed in the network and the innermost hidden layer activated to serve as input for the encoded representation (Fig. 9).

The input data is transformed into a structure where AE stores the utmost significant derived dimensions. It is similar to traditional dimensionality reduction techniques, namely, singular value decomposition (SVD) and principal component analysis (PCA). Deep autoencoder networks can be trained in a greedy manner, which is referred as the stacking process. Some of the autoencoder variants are:

1. Sparse autoencoder (SAE)
2. Variational autoencoder (VAE)
3. Denoising autoencoder (DAE)

6.2.2 Restricted Boltzmann Machine (RBM)

RBM is an unsupervised learning architecture that learns input data representation. It is almost similar to AE; instead, RBMs estimate the probability distribution of the available input data. Hence, it is perceived as a generative model where the data was generated in the underlying process. The canonical RBM is a model that consists of binary visible units $\vec{v}$, and hidden units $\vec{h}$ along with the energy function as shown in the equation:

$$E\left(v,h\right) = -b^T v - c^T v - Wv^T h \tag{21}$$

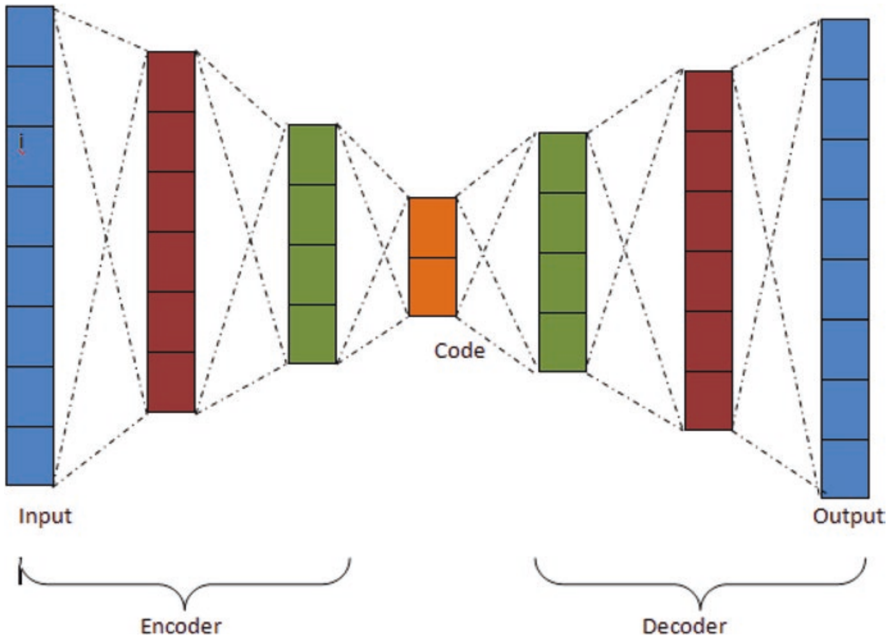

Fig. 9 Autoencoder example and output layers

In Boltzmann machine (BM), every unit is completely connected, while in restricted Boltzmann machine, there is no connection among the hidden units. Restricted Boltzmann machine is typically trained using a stochastic optimization like Gibbs sampling, and it yields the learned representation of the given input data that is viewed as the final form of h. Moreover, RBMs can be stacked hierarchically to construct a deep belief network (DBN) particularly for supervised learning.

7 Platforms for Deep Learning/Deep Learning Frameworks

Many software packages are available for researchers to ease the construction of deep learning architectures, but few years back non-deep learning professionals faced many difficulties to hadle the software packages. This circumstance lasted until Google introduced the DistBelief system in 2012. Following DistBelief, similar software packages like TensorFlow, Microsoft Cognitive Toolkit (previously CNTK), DeepLearning4j, Caffe, Torch, Keras, Neural Designer, H2o.ai, and Deep Learning Kit have extensively spurred the industry (Fig. 10).

Fig. 10 Deep learning platform and output layers

7.1 TensorFlow

The concept of TensorFlow is highly associated with the mathematics involved in engineering and physics. Later TensorFlow has made its way to computer science which is associated with logic and discrete mathematics. Advanced machine learning concepts utilize the manipulation and calculus of tensors. TensorFlow is an open-source end-to-end machine learning library for production and research. It offers APIs for expert and beginner-level learners to develop applications for the cloud, mobile, web, and desktop. For beginner-level learners, TensorFlow recommends Keras API to develop and train the deep learning models. For advanced operations like forward passes, customizing layers, and training the loops with auto-differentiation, define-by-run interface API is recommended. Pre-made estimators are available to implement common machine learning algorithms. The architecture of TensorFlow is divided into four functioning parts, namely, data processor, model builder, training, and estimating the model. It accepts the inputs as tensors or multi-dimensional array, constructs operation flowchart which explains the multiple operations subjected to input, and finally comes out as output. Hence the name TensorFlow arises as the tensors flow through a list of operations and produce the desired output on the other side. TensorFlow is based on static graph computation, to visualize the constructed neural network with the help of TensorBoard. TensorFlow supports algorithms like classification, linear regression, deep learning wipe, deep learning classification, boosted tree classification, and boosted tree regression.