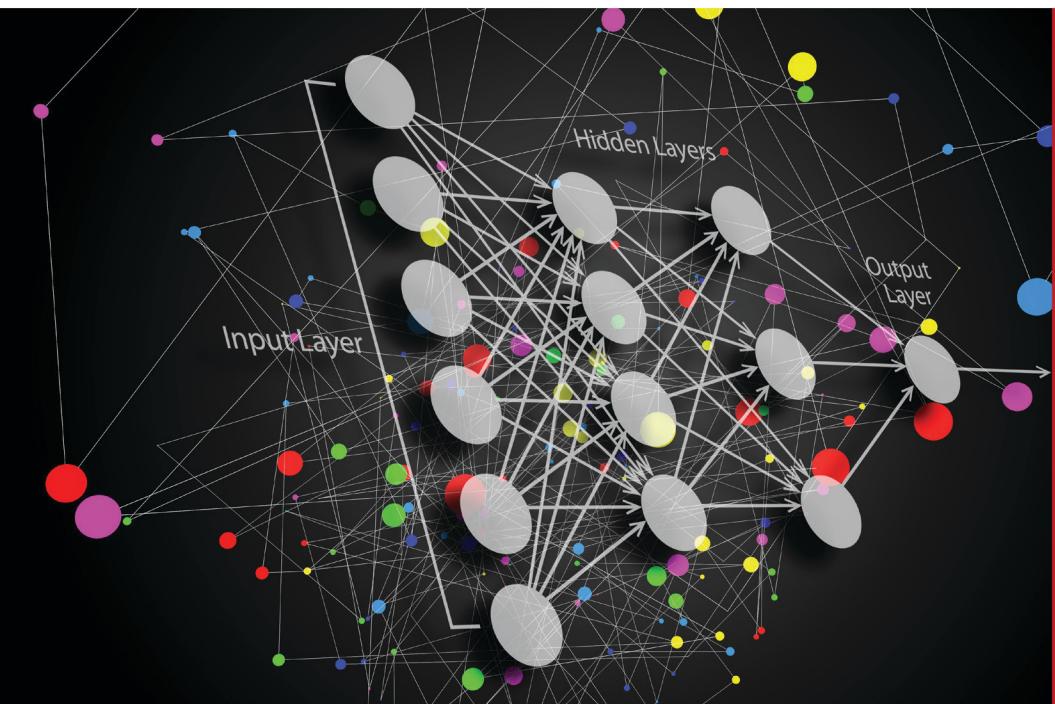


Deep Learning

A Beginners' Guide



Dulani Meedeniya



CRC Press
Taylor & Francis Group

A CHAPMAN & HALL BOOK

Deep Learning

This book focuses on deep learning (DL), which is an important aspect of data science, that includes predictive modeling. DL applications are widely used in domains such as finance, transport, healthcare, automanufacturing, and advertising. The design of the DL models based on artificial neural networks is influenced by the structure and operation of the brain. This book presents a comprehensive resource for those who seek a solid grasp of the techniques in DL.

Key features:

- Provides knowledge on theory and design of state-of-the-art deep learning models for real-world applications.
- Explains the concepts and terminology in problem-solving with deep learning.
- Explores the theoretical basis for major algorithms and approaches in deep learning.
- Discusses the enhancement techniques of deep learning models.
- Identifies the performance evaluation techniques for deep learning models.

Accordingly, the book covers the entire process flow of deep learning by providing awareness of each of the widely used models. This book can be used as a beginners' guide where the user can understand the associated concepts and techniques. This book will be a useful resource for undergraduate and postgraduate students, engineers, and researchers, who are starting to learn the subject of deep learning.



Dulani Meedeniya is a Professor in Computer Science and Engineering at the University of Moratuwa, Sri Lanka. She holds a PhD in Computer Science from the University of St Andrews, United Kingdom. She is the director of the Bio-Health Informatics group at her department and engages in a number of collaborative research projects. She is a co-author of 100+ publications in indexed journals, peer-reviewed conferences, and book chapters. Prof. Dulani has received several awards and grants for her contribution to research. She serves as a reviewer, program committee, and editorial team member

in many international conferences and journals. Her main research interests are deep learning, software modeling and design, bio-health informatics, and technology-enhanced learning. She is a Fellow of HEA (UK), MIET, Senior Member of IEEE, Member of ACM, and a Chartered Engineer registered at EC (UK).



Taylor & Francis
Taylor & Francis Group
<http://taylorandfrancis.com>

Deep Learning

A Beginners' Guide

Dulani Meedeniya



CRC Press

Taylor & Francis Group

Boca Raton London New York

CRC Press is an imprint of the
Taylor & Francis Group, an **informa** business

Designed cover image: Pdusit, Shutterstock Illustrator

First edition published 2024

by CRC Press

6000 Broken Sound Parkway NW, Suite 300, Boca Raton, FL 33487-2742

and by CRC Press

4 Park Square, Milton Park, Abingdon, Oxon, OX14 4RN

CRC Press is an imprint of Taylor & Francis Group, LLC

© 2024 Dulani Meedeniya

Reasonable efforts have been made to publish reliable data and information, but the author and publisher cannot assume responsibility for the validity of all materials or the consequences of their use. The authors and publishers have attempted to trace the copyright holders of all material reproduced in this publication and apologize to copyright holders if permission to publish in this form has not been obtained. If any copyright material has not been acknowledged please write and let us know so we may rectify in any future reprint.

Except as permitted under U.S. Copyright Law, no part of this book may be reprinted, reproduced, transmitted, or utilized in any form by any electronic, mechanical, or other means, now known or hereafter invented, including photocopying, microfilming, and recording, or in any information storage or retrieval system, without written permission from the publishers.

For permission to photocopy or use material electronically from this work, access www.copyright.com or contact the Copyright Clearance Center, Inc. (CCC), 222 Rosewood Drive, Danvers, MA 01923, 978-750-8400. For works that are not available on CCC please contact mpkbookspermissions@tandf.co.uk

Trademark notice: Product or corporate names may be trademarks or registered trademarks and are used only for identification and explanation without intent to infringe.

ISBN: 9781032473246 (hbk)

ISBN: 9781032487960 (pbk)

ISBN: 9781003390824 (ebk)

DOI: 10.1201/9781003390824

Typeset in Times

by Newgen Publishing UK

Contents

Preface.....	ix
Acknowledgements.....	xi
List of Abbreviations.....	xiii
Chapter 1 Introduction	1
1.1 Data-Driven Decision-Making and Society	1
1.2 Overview of Deep Learning.....	2
1.3 Bias and Variance.....	6
1.3.1 Skewness of Data	7
1.3.2 Bias.....	7
1.3.3 Variance	8
1.3.4 Trade-off Between Bias and Variance	8
1.4 Supervised and Unsupervised Learning.....	11
1.5 Supportive Tools and Libraries	12
1.5.1 TensorFlow	13
1.5.2 Keras.....	13
1.5.3 PyTorch	14
1.5.4 Jupyter Notebook	14
1.5.5 NumPy and Pandas.....	14
1.5.6 Tensor Hub	14
Review Questions	15
Chapter 2 Concepts and Terminology	16
2.1 Understanding Neural Networks.....	16
2.2 Regression.....	18
2.2.1 Linear Regression.....	19
2.2.2 Logistic Regression	20
2.2.3 Other Regression Methods	20
2.3 Classification.....	21
2.4 Hyperparameters	22
2.4.1 Overview	22
2.4.2 Weight Initialization	22
2.4.3 Activation Function	24
2.4.4 Learning Rate	29
2.4.5 Loss Function	29
2.4.6 Other Hyperparameters	32
2.5 Model Training.....	33
2.5.1 Model Selection.....	33
2.5.2 Model Convergence.....	33

2.5.3	Overfitting and Underfitting	34
2.5.4	Regularization	37
2.5.5	Network Gradients	38
	Review Questions.....	41
Chapter 3	State-of-the-Art Deep Learning Models: Part I.....	42
3.1	Overview of Neural Networks	42
3.2	Artificial Neural Networks.....	43
3.3	Recurrent Neural Network (RNN).....	45
3.4	Convolutional Neural Networks	48
3.4.1	Overview of Convolutional Neural Network.....	48
3.4.2	Concepts of CNN	49
3.4.3	Convolutional Layer.....	52
3.4.4	Pooling Layer	54
3.4.5	Fully Connected Layer	55
3.5	Comparison of ANN, RNN, and CNN.....	56
	Review Questions.....	58
Chapter 4	State-of-the-Art Deep Learning Models: Part II	59
4.1	Feed-Forward Neural Network	59
4.2	Multi-layer Perceptrons	61
4.3	Generative Adversarial Network (GAN)	62
4.4	Variations of CNNs	64
4.4.1	Residual Networks (ResNet).....	64
4.4.2	Inception Model	66
4.4.3	GoogLeNet	67
4.4.4	Xception Model.....	69
4.4.5	DenseNet Model.....	69
4.4.6	MobileNet Model	70
4.4.7	VGG Model	71
4.4.8	Comparison of CNN Architectures	71
4.5	Capsule Network.....	73
4.6	Autoencoders	76
4.7	Transformers	78
	Review Questions.....	83
Chapter 5	Advanced Learning Techniques	84
5.1	Transfer Learning.....	84
5.1.1	Overview of Transfer Learning	84
5.1.2	Transfer Learning Process	85
5.1.3	Transfer Learning Types, Categories, and Strategies	87
5.1.4	Transfer Learning Applications.....	89
5.1.5	Transfer Learning Challenges	90

5.2	Reinforcement Learning	91
5.2.1	Overview of Reinforcement Learning.....	91
5.2.2	Reinforcement Learning Process.....	91
5.2.3	Implementation and Scheduling Types	94
5.2.4	Applications of Reinforcement Learning.....	95
5.2.5	Challenges of Reinforcement Learning.....	95
5.3	Federated Learning	96
5.3.1	Overview of Federated Learning.....	96
5.3.2	Federated Learning Process.....	97
5.3.3	Types and Properties of Federated Learning	100
5.3.4	Applications of Federated Learning	101
5.3.5	Challenges of Federated Learning.....	102
5.4	Multi-modeling with Ensemble Learning.....	103
5.4.1	Overview of Ensemble Learning.....	103
5.4.2	Ensemble Learning Process.....	103
5.4.3	Ensemble Learning Techniques.....	106
5.4.4	Applications of Ensemble Learning	110
	Review Questions	110
Chapter 6	Enhancement of Deep Learning Architectures.....	112
6.1	Model Performance Improvement	112
6.2	Regularization	115
6.3	Augmentation.....	119
6.4	Normalization	120
6.5	Hyperparameter Tuning	123
6.6	Model Optimization	125
6.6.1	Overview of Model Optimization	125
6.6.2	Gradient-Based Optimization Algorithms.....	127
6.6.3	Other Optimization Algorithms.....	130
6.7	Neural Architecture Search (NAS)	132
6.7.1	Overview of NAS	132
6.7.2	NAS Process.....	133
6.7.3	Search Space.....	134
6.7.4	Search Strategies of NAS	136
6.7.5	Strategies for Performance Measures.....	139
6.8	Adversarial Training	140
6.8.1	Overview of Adversarial Training	140
6.8.2	Types of Adversarial Attacks.....	141
6.8.3	Adversarial Attack Generation Techniques	142
6.8.4	Adversarial Attack Defensive Methods.....	144
6.8.5	Best Practices to Avoid Adversarial Attacks	145
	Review Questions	145

Chapter 7	Performance Evaluation Techniques	147
7.1	Overview of Performance Measures.....	147
7.2	Types of Performance Metrics	148
7.2.1	Confusion Matrix	148
7.2.2	Accuracy	148
7.2.3	Precision and Recall	150
7.2.4	F-Measure.....	151
7.2.5	Specificity and Sensitivity	152
7.2.6	Receiving Operating Characteristic Curve (ROC)	152
7.2.7	Area Under the ROC Curve (AUROC) and AUC	153
7.2.8	Cross-Validation	153
7.2.9	Kappa Score	157
7.2.10	Grad-CAM Heat Map.....	157
7.2.11	Metrics for Imbalanced Datasets.....	158
7.2.12	Metrics for Regression Problems	159
7.2.13	Summary of Performance Metrics	163
	Review Questions	163
	Appendix – Frequently Asked Questions	165
	References	173
	Index	181

Preface

The rapid development of digital technologies has resulted in an explosive growth of data. Data engineering plays an essential role in many fields, including finance, medical informatics, and social sciences. This has led to increasing demand for career opportunities with the knowledge and experience of data science, with competence in computer programming. However, still, there is a global shortage of workforce whose skills span these areas.

Deep learning (DL) is an important and evolving area in data science that includes statistics and predictive modeling. It is concerned with algorithms inspired by the brain's structure and functions known as artificial neural networks. DL can automatically learn features in data, by updating learned weights at each layer. This book provides adequate theoretical coverage of DL techniques and applications. This book will teach deep learning concepts from scratch. We aim to make DL approachable by teaching the concepts and theories behind DL models. Thus, practitioners can grab the critical thinking skills required to formulate problems, design and develop models to make accurate predictions and support the decision-making process. Many academic institutions have embarked on DL education and research at various levels. At present, DL has become a forward-looking academic discipline with a wide range of real-world applications.

DL is extremely beneficial to data scientists in collecting, analyzing, and interpreting large amounts of data with efficient processing. There are many advantages associated with deep learning. For instance, DL techniques may produce new features from a small collection of features in the training dataset without any further human interaction. The ability to process large numbers of features makes DL techniques very powerful when dealing with unstructured data namely texts, images, and voices. More reliable and concise analysis results can be obtained as the prediction process is based on historical data. In the long run, it also supports improving prediction accuracy by learning from flaws. Although DL techniques can be expensive to train, once trained, it is cost-effective. Moreover, these techniques are scalable, as they can analyze large volumes of data and execute numerous calculations in a cost- and time-effective way.

DL applications are widely used in several industries like finance, transport, healthcare, automanufacturing, and advertising. For instance, DL is reshaping and enhancing the living environments by delivering new possibilities to improve people's life. For instance, in the healthcare domain, it helps in the early detection of cancer cells and tumors, improves the time-consuming process of synthesizing new drugs, and invents sophisticated medical instruments. Deep learning is used in the entertainment industry such as Netflix, Amazon, and Film Making. Netflix and Amazon use recommender systems to provide a personalized experience to their viewers using their show preferences, time of access, and history. Voice and audio recognition technology can be used to train a deep learning network to produce music compositions. Google's Wavenet and Baidu's Deep Speech can train a computer to learn the patterns

and the statistics that are unique to the music. It can then generate a completely new composition. Additionally, the ‘LipNet’, which is developed by Oxford and Google scientists, could read people’s lips with 93% success. This can be used to add sounds to silent movies. Further, in advertising, DL allows optimizing a user’s experience. Deep learning helps publishers and advertisers to increase the relevance of the ads and boost the advertising campaigns.

Acknowledgements

We are grateful to all who helped improve the content and offered valuable feedback. Specifically, we thank K. T. S. De Silva and S. Dayarathna and T. Shyamalee for their contributions to collecting materials and designing graphics.



Taylor & Francis
Taylor & Francis Group
<http://taylorandfrancis.com>

Abbreviations

Adaptive delta	(AdaDelta)
Adaptive gradient	(Adagrad)
Adaptive moment estimation	(ADAM)
Area under the curve	(AUC)
Area under the ROC curve	(AUROC)
Artificial intelligence	(AI)
Artificial neural network	(ANN)
Bi-directional encoder representations from transformers	(BIRT)
Capsule network	(CapsNet)
Carlini & Wagner attack	(C&W)
Convolutional neural networks	(CNN)
Deep learning	(DL)
Deep neural network	(DNN)
Differentiable architecture search	(DART)
Directed acyclic graph	(DAG)
Efficient neural architecture search	(ENAS)
Exponential moving average	(EMA)
Facebook-Berkeley-Nets	(FBNet)
False negative	(FN)
False positive	(FP)
Fast and practical neural architecture search	(FPNAS)
Fast gradient sign method	(FGSM)
Federated learning	(FL)
Generative adversarial networks	(GAN)
Geometric Mean	(G-mean)
Gradient descent	(GD)
Gradient-weighted class activation mapping	(Grad-CAM)
Internet-of-things	(IoT)
Jacobian-based saliency map attack	(JSMA)
Limited-memory Broyden-Fletcher-Goldfarb-Shanno	(L-BFGS)
Logarithmic loss	(Log loss)
Long short-term memory networks	(LSTM)
Machine learning	(ML)
Matthew's correlation coefficient	(MCC)
Mean absolute error	(MAE)
Mean squared error	(MSE)
Multilayer perceptron	(MLP)
Neural architecture optimization	(NAO)
Neural architecture search	(NAS)
Natural language processing	(NLP)

Peer-to-peer	(P2P)
Principal component analysis	(PCA)
Receiver operating characteristics	(ROC)
Rectified linear	(ReLU)
Recurrent neural networks	(RNN)
Region of interest	(ROI)
Residual network	(ResNet)
Root mean square error	(RMSE)
Root mean square propagation	(RMSprop)
Stochastic gradient descent	(SGD)
Stochastic neural architecture search	(SNAS)
True negative	(TN)
True positive	(TP)
Vision transformer	(ViT)
Visual geometric group	(VGG)
Youden's index	(YI)

1 Introduction

1.1 DATA-DRIVEN DECISION-MAKING AND SOCIETY

In a world where data-driven decision-making is becoming increasingly common, machine learning and artificial intelligence have come to be seen as valuable resources for making better and faster decisions. The advanced technological developments in the field of deep learning, which is a specialization of machine learning, have produced powerful tools for a wide range of applications. Advances in these fields have enabled computers to extract features, detect patterns, and make predictions about data and outcomes, with possible explanations to increase the trustworthiness of the applications. As a result, these techniques are being increasingly used in a wide variety of fields, including healthcare, education, finance, and social.

With the development of devices that generate large piles of data, we encountered a new concept of big data in the past decade. New analytic methodologies and data collection platforms have been created based on this concept and today we are exposed to a massive amount of data in every possible area of interest. The moment you are reading this, there are thousands of internet-of-things (IoT) devices, your mobile phones, or any other ubiquitous device that generates and sends data throughout the world among different networks.

Let us move our topic of discussion into data-driven decision-making, which is a common phenomenon found in the technical field, which provides new avenues to explore the usages in the decision-making process applied to different scenarios in various domains. For example, if we dive into the decision-making process powered by data in the business world, multiple causes can be used to describe the initiative. First, the collection of survey responses by the stakeholders of the business can be used to identify the enhancements or properties of their products, services, or features of their customers. Furthermore, with the use of advanced analytics, predictions can be made on how they are going to develop these new features to adhere to customers' liking. Also, some tests can be used as user testing to monitor their customers in using their products or services and enable them to identify potential issues associated with the development before the official release, which will lead to enhanced customer satisfaction and lesser defects or bugs.

From another point of view, when launching a brand-new product or to the market, data-driven decision-making enables us to analyze the patterns and behaviors in that

market and understand how the new product is going to perform in that particular market. This enables increased efficiency in market research and can immensely support many departments, including marketing, upper management, and even the development level to get clarity in the decision-making process. Most importantly, analyzing the patterns and shifts or new tendencies in the market based on demographic data applies to all sorts of businesses and different industries. Therefore, the data-driven decision-making process immensely supports organizations to determine opportunities or potential threats, where the organization can prepare with potential ways to tackle them.

In the applications of such data-driven decision-making, artificial intelligence (AI) takes a significant role. Since most of the activities are carried out by computers, the decision-making process is also supported by computational implementations more efficiently and effectively than human intervention. Developing artificial intelligence-enabled applications has been a topic of interest for a couple of decades. From email spam filters to autonomous cars, AI supports a wide range of applications, which are used to guide the human thinking process. In the early stages of artificial intelligence, knowledge was used to solve problems that were difficult to solve with human intelligence.

The usage of artificial intelligence to assist in decision-making will depend on a collection of factors including the current issues, vision, goals, nature of the application, and the type and quality of data to which it is exposed. This has become an essential tool to assist in making smarter and more impactful decisions. Therefore, decision-making is hugely benefited from data which is powered by the usage of advanced AI methodologies including machine learning and deep learning.

1.2 OVERVIEW OF DEEP LEARNING

Have you ever wondered how deep learning evolved from machine learning? This is understandable by comparing the differences between machine learning and deep learning. The improvements in computational technologies try to simulate the human intelligence process using machines. Initially, let us understand the concepts behind artificial intelligence (AI), machine learning (ML), and deep learning (DL) as shown in Figure 1.1.

- Artificial intelligence (AI) has the ability to perform tasks using machines that normally require human intelligence. It can be considered as a smart application that simulates the behavioral patterns of humans and learns without human intervention such as self-driving cars.
- Machine learning (ML) can be considered as a specialization of AI that consists of a stack of tools to analyze and visualize data, and predictions. It can learn using data without being explicitly programmed with a set of rules. This approach is based on training a model from datasets.
- Deep learning (DL) is a type of ML that simulates the human brain. It uses multiple layers in a deep neural network to progressively extract high-level features from the raw input. Their ability to analyze more complex relationships makes them particularly useful for modeling a wide variety of real-world problems.

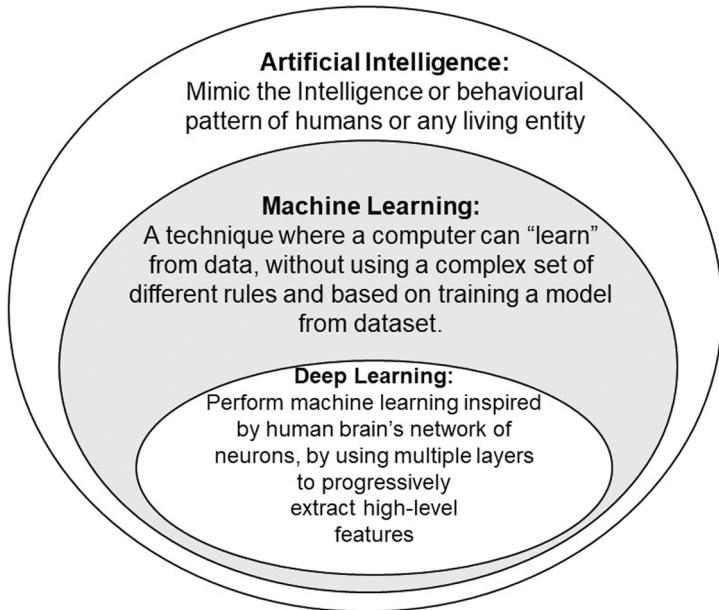


FIGURE 1.1 Overview of AI, ML, and DL.

Deep learning can be considered an evolution of machine learning. Earlier, machine learning algorithms were used to develop models for different applications. Many machine learning algorithms were developed to learn data and improved over time and are still used for making intelligent decisions. Deep learning is a subfield of machine learning with a multi-layered neural network that can learn and make intelligent decisions on its own. Deep learning models can learn the high-level features from the data on their own, while machine learning models need manually engineered features that are identified by domain experts. The evolution of these technologies is shown in Figure 1.2. Although the concept of deep learning was theorized earlier, it became more popular among data scientists recently. The main reasons can be stated as the recent advent of big data and high computational power GPUs. Most of the computationally infeasible algorithms and models became feasible technologically and concept-wise with the availability of a large amount of data, inexpensive data storage, and computation power. Consequently, deep learning models are widely used to solve real-world problems involving tasks like image recognition, speech recognition, and natural language processing.

Deep learning is defined as a family of machine learning models that are characterized by their depth and generality. These models can learn complex, non-linear relationships between input data elements and the corresponding output. The implementation of these models uses artificial neural networks with multiple hidden layers; hence defined as deep neural networks (DNNs). While the data is transformed through these multiple hidden layers, each level learns the input data and transforms it into a slightly abstract and composite representation and eventually captures

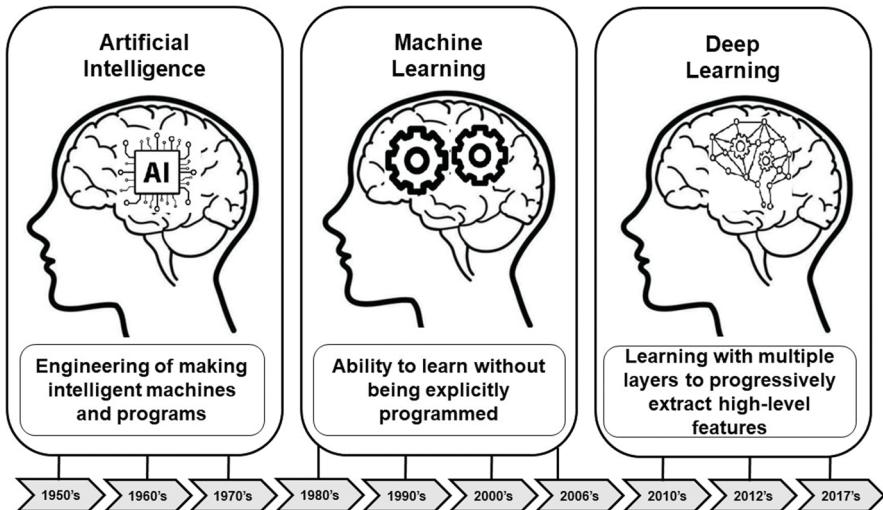


FIGURE 1.2 Evolution of technologies.

complicated relationships. Consider the image recognition application shown in Figure 1.3. For instance, the input can be a set of pixels in a matrix. The first layer may extract the pixels related to the edges. The second layer may compose the set of edges. The third layer may arrange the edges into different shapes. The fourth layer may predict the image. Likewise, the deep learning process extracts the features that can be learned at different levels. During this process, parameter tuning is needed to obtain optimal results by changing the number of layers and the size of the layers. Therefore, deep learning learns progressively to extract features and make optimal predictions.

Overall, in a DNN data flows in a feed-forward direction from the input to the output layer. A neural network consists of mainly three types of layers namely, input, hidden, and output layers. Initially, the DNN creates the linkages between neurons and assigns random numerical weights for the connections. In this process, the weight values and inputs are multiplied and generate an output within the range 0 and 1. The algorithm adjusts the weights until the expected prediction accuracy is reached. Subsequently, the algorithm makes several parameters more prominent in hidden layers, until the optimal equations are obtained to process all the data.

Considering the data transformation between inputs and outputs, machine learning needs the necessary representation of the data that is suitable for algorithms to transform into output. However, deep learning models learn many layers of transformation while each layer offers a representation at one level. For example, layers that are near the input contain fewer details of the data while layers that are near the output have high-level data representation with the concepts used for the data discrimination. The deep learning models can be identified as multi-level representation learning. These models with many layers are more capable of extracting low-level perceptual data than other models. This not only performs better than other shallow models, but it

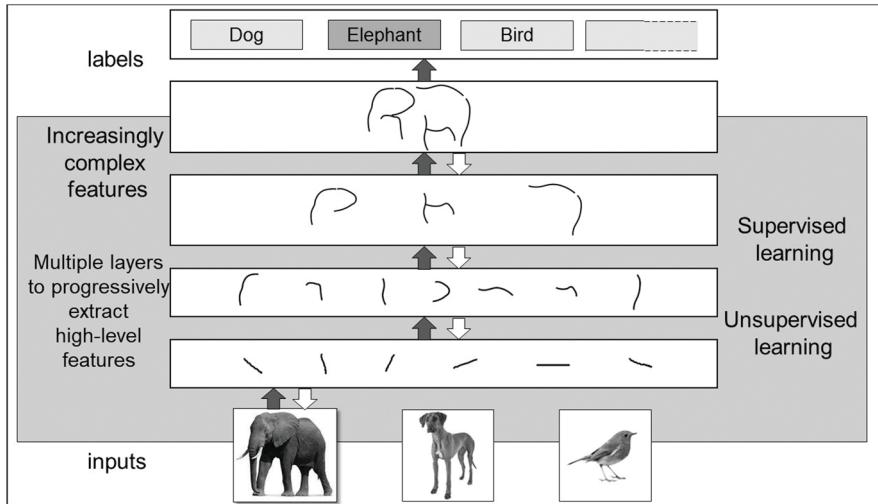


FIGURE 1.3 Overall process within a deep learning model.

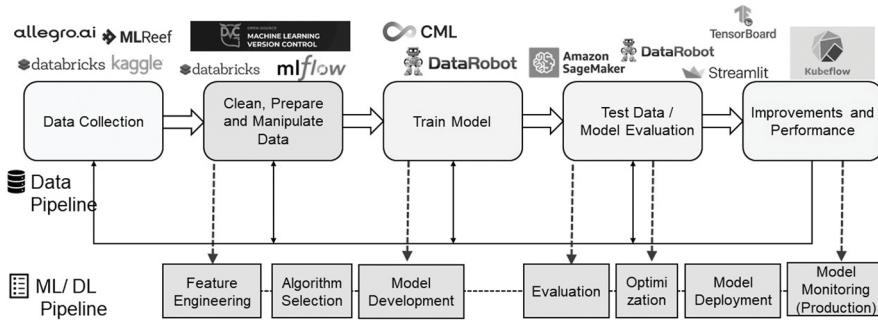


FIGURE 1.4 Life cycle of a deep learning process.

also has an accurate and automated feature engineering process by eliminating many boundaries in areas such as natural language processing, computer vision, and speech recognition.

With the overview of the deep learning concept, it is important to identify the data processing stages to produce insights and predictions to obtain the outcome in practice. As shown in Figure 1.4, the life cycle of a deep learning process mainly consists of data acquisition, preprocessing, training, testing, evaluation, deployment, and monitoring. We learnt that the overall effectiveness of the model depends mainly on the data. Generally, data can be private or public and collected using surveys or experiments. With the availability of data, a data scientist needs to understand the data by exploring the structure, relevance, type, and suitability of data. Although data preparation is a time-consuming task, it plays an important role in the life cycle, to derive new features from the existing data. This, exploratory data analysis is required

to identify the affecting factors using the data distribution between different feature variables before the actual model design. Accordingly, in the data flow pipeline, the data should be preprocessed to clean, remove outliers, manage missing data, normalize, and augment before feeding into the training model. Considering the deep learning pipeline, the feature engineering techniques are aligned with the data preprocessing to extract and identify informative features of the data.

The training algorithm can be selected considering different factors such as the type of data, nature of the application, and resource availability. The model training and testing processes are engaged in tuning the hyperparameter and applying optimizations to generate better results. These models should be designed to learn the data and perform well on new data as well, by ensuring the balance between performance and generalizability. Once the model is evaluated by testing on unseen data. The modeling process should be reiterated until the desired level of metrics is achieved. A detailed description of these concepts and techniques is discussed in later chapters of this book. Once the final model is deployed, the application is monitored for further improvements. In practice, several frameworks and technologies are available to ease the processes in the deep learning life cycle. Furthermore, different tools and frameworks are utilized to accomplish these processes.

Deep learning is still evolving with novel ideas of big data processing with artificial intelligence. Therefore, you need to better understand deep learning techniques and their key concepts for the development of innovative applications. This book will provide you with the theoretical background on basic deep learning techniques, neural networks, deep learning models, types of deep learning approaches, architectural enhancements, and evaluation techniques.

1.3 BIAS AND VARIANCE

In general, a machine learning algorithm aims to correctly determine the mapping function to predict a variable y (output) given x (input). However, there is always a difference between model predictions and actual results, which is known as prediction error. Therefore, it is required to have an awareness of the bias and the variance errors, when training a machine learning model. These fundamental concepts on parameter estimation, bias, and variance are useful in identifying model characteristics on generalization, overfitting, underfitting, and accurate predictions.

Let us start with the basics of point estimation and interval estimation with simple statistics. Point estimation calculates a single value of an unknown parameter such as a model weight or a whole function. Since it estimates the relationship between the input and output, it is also known as a function estimator. For example, the sample mean is considered as a point estimation. The interval estimation results in a range of values that a parameter can remain. For example, the confidence interval is considered as an interval estimation.

Let us denote the point estimation of a parameter θ by $\hat{\theta}$ to differentiate the estimation of parameters from their actual values. Let $\{X_1, X_2, \dots, X_m\}$ be an independent and distributed set of data points. The point estimator can be defined as a function of data, $\hat{\theta} = f\{X_1, X_2, \dots, X_m\}$, where ' f ' returns a value close to the actual value of θ .

In general, a function, where the predicted output is closer to the actual value of θ , is considered as a good estimator. It is important to review the properties of these estimators. The following sections describe the bias and variance. A correct balance between bias and variance is important to generate accurate predictions.

1.3.1 SKEWNESS OF DATA

Skewness determines how far a random variable's probability distribution deviates from the normal distribution (probability distribution without any skewness), as shown in Figure 1.5. Also, skewness indicates the direction of outliers.

For example, let us consider the case of positive skew data, where a large number of data instances consist of small values. This results in better training performance at predicting data instances with lower values. Thus, there is a 'bias' towards lower values, in this scenario. Considering the direction in this case, most of the outliers appear on the right side of the distribution. Thus, there is a variance in data.

1.3.2 BIAS

The term bias can be defined as the deviation between the predicted value by the deep learning model and the actual output or the ground truth. When the bias is a higher value, it indicates a large error in the output of the model. Also, it can indicate the imbalance of the dataset. Generally, we expect a model to have a low bias to prevent issues such as the underfitting of data to the model. This can be explained as a systematic error of the training model, which skews the result in favor or against the actual output. Bias shows the matching of the dataset to the model as follows:

In a high-bias situation, the dataset does not match the model.

In a low-bias situation, the dataset fits with the training model.

The following indicators help to identify a high-bias model:

- Failure to capture the data trends
- Potential towards underfitting
- More generalized/overly simplified
- High error rate

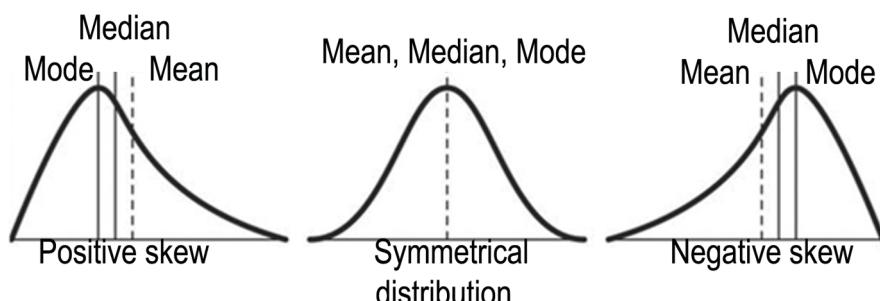


FIGURE 1.5 Data distributions.

The bias of an estimator can be defined as in (1.1), where $\hat{\theta}$ denotes the estimation of a parameter where the actual value is θ , and $\hat{\theta}m$ is the point estimator. The term $E(\hat{\theta})$ is an expectation of the data. If $\text{Bias } (\hat{\theta}) = 0$, the estimator of $\hat{\theta}$ is considered as unbiased, as $E(\hat{\theta})$ is equal to θ .

$$\text{Bias } (\hat{\theta}) = E (\hat{\theta}) - \theta \quad (1.1)$$

1.3.3 VARIANCE

Variance indicates the expected difference between the observed data instances from the average value. Thus, variance indicates the data spreading within the sample set. Both bias and variance of an estimator are calculated for a dataset. This variance measures how the estimate would vary as the dataset is changed independently from the underlying data generating process. In other words, it indicates the changes in the model with different parts of the training data set.

Since the target function is estimated from the dataset, it is acceptable to have some variance. However, it should not vary drastically from one dataset to another, which indicates that the estimator is good at understanding the hidden mapping between inputs and outputs. It can be considered as an indicator of the uncertainty in the data. A high variance indicates that the estimator does not generalize on unseen training datasets. In that case, the model shows high performances on the training set but gives high error rates on the testing set. It is good to have a relatively low variance for an estimator.

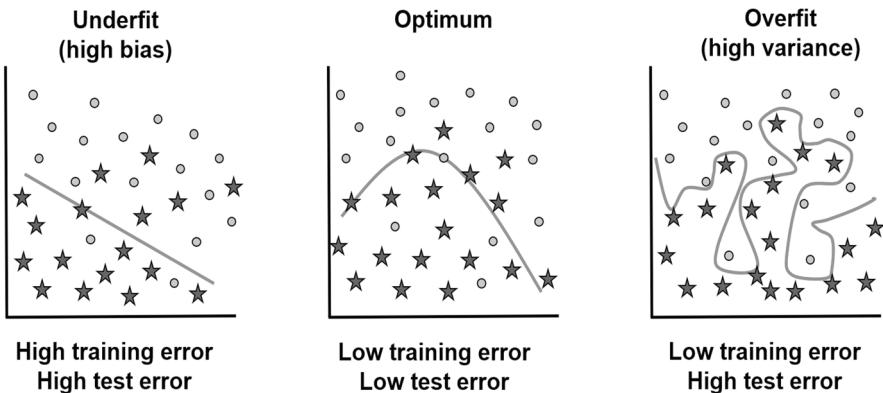
The following indicators help to identify a high-variance model:

- Noise in the dataset
- Potential towards overfitting
- Complex models
- Trying to include all data points closer.

1.3.4 TRADE-OFF BETWEEN BIAS AND VARIANCE

As we already discussed, bias and variance are used to show the errors in an estimator. Overall, bias and variance calculate the difference from the actual value, and the deviation from the expected estimator with the changes in the dataset, respectively, during model training and testing. It is expected to have a balance between these terms, that is, low bias and low variance. Figure 1.6 visually explains the bias and variance in feature classification of 2D space. A detailed description is included in Chapter 2. Additionally, Table 1.1 states a comparison of bias and variance, and Table 1.2 shows the trade-off between bias and variance with different training and testing error values.

Accordingly, if the model has a high bias, it has made more assumptions about the target function. Underfitting may result from missing significant relationships between characteristics and outputs. The changes in training data will produce substantially diverse target functions if a model has a high variance. As a result, the model overfits and starts to learn the random noise instead of the output. Since the model has a greater capacity to learn from the training data, increasing model complexity would

**FIGURE 1.6** Bias and variance of a classifier.**TABLE 1.1**
Comparison of Bias and Variance

	Bias	Variance
High model complexity	Low bias	High variance
Causes	High bias results in underfitting	High variance results in overfitting.
Feature	Low bias indicates fewer target function assumptions are made.	Low variance means that similar target functions would result from training data changes.

TABLE 1.2
Trade-off Between Bias and Variance

Train error	Very low	Relatively high	Relatively high	Very low
Test error	High	Relatively high	Very high	Low
Bias-variance	High variance	High bias	High bias and high variance	Low bias and low variance

typically result in a decrease in bias error. The variance error will rise as a result, though, and the model may start to pick up on noise in the training set.

However, since bias and variance are inversely connected, it is hard to have a model with a low bias and a low variance. Therefore, the trade-off between these terms can be stated as follows.

High-bias models will have low variance.

High-variance models will have a low bias.

Generally, when the model is simple with few parameters then it may have high bias and low variance. Here, the model may not have the risk of generating inaccurate results, but it will not match the dataset. In contrast, if the model has many parameters, then it will have high variance and low bias. In this situation, although the model fits with the dataset, there are more chances to predict inaccurate results. These aspects indicate the model's flexibility to obtain an optimal model. For example, if the model does not fit with the dataset, it will have a high bias. This leads to an inflexible model with low variance.

This trade-off can be addressed, and the most suitable model can be selected by comparing the mean squared error (MSE) of the estimators as in (1.2). The estimators with less MSE can keep both their bias and variance in an acceptable range. It should be noted that these biases and variances are linked with capacity, overfitting, and underfitting concepts in deep learning.

$$\text{MSE} = E [(\hat{\theta} - \theta)^2] = \text{Bias} (\hat{\theta})^2 + \text{Var} (\hat{\theta}) \quad (1.2)$$

Consider, Figure 1.7 with model capacity against the error. The capacity of the model indicates its capability to suit a variety of functions. When the capacity increases, the bias of the model tends to decrease, and variance gets increased. This produces another U-shaped curve, which represents the generalization error. As capacity varies, there is an optimal point in the graph that denotes a good balance between bias and variance that minimizes the error. However, a learning algorithm can handle some variance. A model that is optimally balanced between bias and variance is neither overfitting nor underfitting.

A trained model with the lowest bias versus variance trade-off for a specific dataset is the desired outcome. Techniques such as cross-validation, regularization, dimension reduction, stop training early and use mode data will help overcome bias and

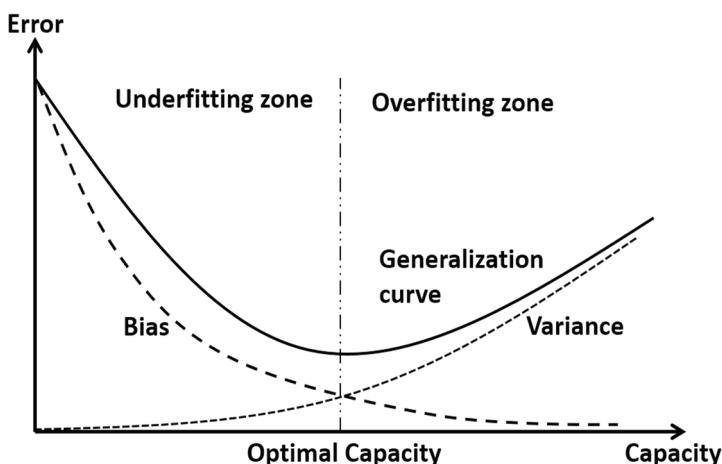


FIGURE 1.7 Model error variation with respect to model capacity.

variance errors. The following tasks can be applied to address the trade-off between bias and variance.

- Increase the complexity of the model. This decreases the overall bias while increasing the variance to an acceptable level. This aligns the model with the training dataset without incurring significant variance errors.
- Increase the training dataset. This is the preferred method when dealing with overfitting models. This allows users to increase the complexity without variance errors that negatively impact the model with a large dataset. A learning algorithm can be generalized easily when there are many data points. However, when the data is underfitting or the model shows low bias, the model is not sensitive to the dataset, even in a large dataset. Therefore, for models with high bias and high variance, using a large dataset is a feasible solution.

1.4 SUPERVISED AND UNSUPERVISED LEARNING

The real-life problems can be categorized into two main sets, that is, yes or no questions and quantification questions. The yes or no question often requires binary decision-making procedures, where the evidence or data is presented. The quantification problems require data and calculations to find the quantified answers. The same approach is taken in using computers to answer these questions. However, as we see in real-life problem solving, these two types of questions follow two distinctions in the methodology using computers. The problem-solving process with yes or no answers is known as classification and the other type of problem-solving is known as regression to predict a quantified amount. Therefore, these two approaches follow different sets of machine learning algorithms to answer the questions we address. The classification problems can be further divided into supervised and unsupervised learning.

Supervised learning predicts the label of a given data item, given all the resources to navigate through the problem. Here, the dataset needs to be annotated into the expected outcome or classes of the model. If we incorporate supervised learning into the actual learning process, imagine that you are given all the problems, and sample questions and the teacher is teaching you the things that you need to follow to understand the problem and design a solution. In machine learning terminology, for each data point, the expected answer is named as the label. The sample data associated with the label are called features. The goal of supervised learning is to produce a mapping function that maps the features or input data into the label. The mapping function can differ based on the problem. Let us consider a real-life example to understand this terminology. Consider a situation where a doctor needs to predict whether a patient is going to have a stroke after examining and based on his past patient history. In this situation, the patient's history may contain information about previous strokes and their causes, which are the labeled examples provided. The current observation is the input data and the outcome should be whether the person is going to have a stroke or not. Here, the doctor is the decision-maker. However, in a machine learning context, the computer provides this decision using the provided labeled data.

Some of the real-world examples that use supervised learning are listed as follows.

- Predict cancer vs non-cancer given computer tomography images.
- Identify fraud and non-fraud signatures in financial documents.
- Predict the stock prices for the next month based on this month's financial data.
- Identify spam and non-spam emails.
- Classification of positive and negative sentiments from tweets.

In contrast, unsupervised learning categorizes unlabeled datasets, by identifying the hidden patterns and extracting useful information without human intervention. Here, the number of questions and answers solely depends on the quality and the hidden information in the dataset. Let us understand this concept using a real-world example. Consider a supermarket that wants to group its customers, to recommend products. They can apply a data-driven approach, by grouping the customers based on their age group and then deriving insights from these groups.

Some of the examples of unsupervised learning are listed as follows.

- Group a set of random photos into landscape photos, pictures of dogs and cats, babies and mountain peaks, etc. This is known as clustering in machine learning terminology.
- Find a group of small numbers of parameters that can be used to explain the data. This extracts the most important features from the dataset, which explain the dataset the best. This procedure is known as principal component analysis in machine learning terminology.
- Identify the functional proteins that affect the most in cancer diagnosis.
- Identify the patterns in financial fraud activities.
- Identify the important minimum set of dimensions in magnetic resonance imaging data.

1.5 SUPPORTIVE TOOLS AND LIBRARIES

A complete platform to execute deep learning models can be created using a set of programming languages, machine learning libraries, services and web applications. Figure 1.8 shows a set of tool stack that supports building the deep learning models efficiently in terms of resource utilization, maintenance, team efforts, and user experience. We discuss some of the platforms that support deep learning.



FIGURE 1.8 Tool stack to support the deep learning process.

1.5.1 TENSORFLOW

TensorFlow is an open-source platform for fine-tuning large-scale machine learning applications using different libraries, tools, and resources. Initially, this framework was developed by Google for their internal usage and later provided as an end-to-end machine learning platform in the public domain. Among several functionalities, it mainly supports model training and inference of deep neural networks. Since there are a large amount of data to process using complex algorithms, it is required to store data compactly and feed it to the neural network. Tensors provide a better way to represent data as an n -dimensional matrix or a vector. Since these tensors hold data in different known shapes, the shape of the data can be identified with the dimension of the matrix. After storing data in tensors, the relevant computations can be performed in the form of a graph. These tensors can be derived either from input data or as computation results of an operation that conducts inside the graph. The input data goes into the graph at one end and then flows through various operations and comes out at another end as an output. All these operations in the graph are known as graph op nodes that are connected by tensors as edges. These graph frameworks can run on multiple CPUs, GPUs, or mobile operating systems. Accordingly, some of the benefits of using TensorFlow can be stated as open-source, platform independence, train on CPU and GPU, high flexibility, autodifferentiation and manage threads and asynchronous computation.

1.5.2 KERAS

Keras is an open-source deep learning API written in python that executes on machine learning platforms. This provides an interface to solve complex learning problems aiming at deep learning techniques. This API acts as a high-level wrapper to create deep learning models, define their layers, and compile models. However, this does not support other low-level API such as generating computational graphs and making tensors and sessions. Keras supports multiple backends for the computation such as TensorFlow, Theano, CNTK, and PlaidML. TensorFlow uses Keras as its official high-level API, supporting many in-built modules to compute neural networks.

Keras offers a simple API that reduces the complexity of making neural network models and allows you to implement the codes with a simple set of functions. Since Keras supports multiple cross-platforms, a given backend can be selected depending on the requirements. When using TensorFlow with Keras API, we can easily create customized workflows based on the requirements. Also, this is much easier to learn as it provides a python frontend with a high level of abstraction. Keras can be deployed on devices like iOS, Android, Raspberry Pi, Cloud Engines, or Web Browsers with .js support. Also, Keras runs on both GPU and CPU, with the support of in-built data parallelism to process large data volumes for model training. Therefore, this can be used easily and efficiently as a high-end API to create deep learning networks.

Let us learn the main steps in creating a simple Keras model. The basic elements of Keras are models and layers. Initially, we need to define a network by adding layers to support data flow in the selected model type. There are two types of models namely sequential and functional. Then we need to define the loss function, optimizer, and

the other matrices to calculate the model accuracy, and compile the model to convert it into a machine-understandable format. Next, the model can train, evaluate, and predict the results. Additionally, the Keras functional API can be used to build arbitrary graphs of layers or develop models in complex architectures.

1.5.3 PYTORCH

PyTorch is a python-based open-source machine learning framework. It uses an optimized tensor library for deep learning using GPUs and CPUs. One of the main high-level features of PyTorch is its dynamic computational graph based on automatic differentiation. In contrast to TensorFlow, where we need to first define the entire computational graph before running the model, PyTorch allows us to define the graph dynamically. The PyTorch library is designed for more efficient use by tracking the model built in real-time. Since the developers can dynamically change the behavior of the graph, it is easier to use than TensorFlow. Also, PyTorch enables GPU-accelerated tensor computations and effective data parallelism. However, compared to TensorFlow, PyTorch provides limited visualizations during the training process.

1.5.4 JUPYTER NOTEBOOK

Jupyter Notebook is an open-source web application. Developers use this environment to create and share documents with source codes, text, and visualizations. This helps to perform end-to-end workflows in data science such as data preprocessing, model building, model training, data visualization, and many other related works. Jupyter notebooks can use to write codes in independent cells and execute them individually. Therefore, this allows testing specific blocks of code without executing the entire script of code as in many other IDEs. This is a flexible and interactive platform that is widely used in data science.

1.5.5 NUMPY AND PANDAS

The libraries NumPy (Numerical Python) and Pandas (Panel Data) are important in deep learning due to their matrix computation capabilities. Both are open-source Python libraries. NumPy consists of multi-dimensional array objects and a set of procedures to process them for numerical computations. Thus, it supports processing large matrixes using mathematical functions. Pandas is built on top of the NumPy package and supports functionalities such as loading, manipulating, preparing, modeling, and evaluating tasks for multi-dimensional data. NumPy and Pandas modules are best suited for numerical and tabular data, respectively.

1.5.6 TENSOR HUB

TensorFlow hub provides a repository of pretrained models as off-the-shelf models to be used in machine learning tasks. These models are used for fine-tuning to build

real-world applications and learning purposes with few lines of code. The models in this repository support a variety of applications, such as pattern recognition, object detection, audio processing, and natural language processing.

REVIEW QUESTIONS

1. What are the advantages of using deep learning based applications in the real world?
2. Explain the importance of balancing bias and variance.
3. What are the problems of having high-dimensional data and explain possible approaches to address those issues?
4. What aspects need to be considered when selecting the correct support tool or support library when solving a learning problem?

References

1. Abdel-Jaber, H., Devassy, D., Al Salam, A., Hidaytallah, L., El-Amir, M., 2022. A review of deep learning algorithms and their applications in healthcare. *Algorithms* 15, 71. doi: 10.3390/a15020071.
2. Abeysinghe, C., Perera, I., Meedeniya, D., 2021. Capsule networks for character recognition in low resource languages, in: Malarvel, M., Nayak, S.R., Pattnaik, P.K., Panda, S.N. (Eds.), *Machine vision inspection systems, Volume 2: Machine Learning-Based Approaches*. John Wiley and Sons. chapter 2, pp. 23–46. doi: 10.1002/9781119786122.ch2.
3. Agarwal, N., Sondhi, A., Chopra, K., Singh, G., 2021. Transfer learning: Survey and classification . *Smart innovations in communication and computational sciences*, 145–155. doi: 10.1007/978-981-15-5345-513.
- Agarwal, V., Lohani, M., Bist, A.S., Harahap, E.P., Khoirunisa, A., 2022. Analysis of deep learning techniques for chest x-ray classification in context of covid-19. *ADI Journal on Recent Innovation* 3, 208–216. doi: 10.34306/ajri.v3i2.659.
- Al Husaini, M.A.S., Habaebi, M.H., Gunawan, T.S., Islam, M.R., Elsheikh, E.A., Suliman, F., 2022. Thermal-based early breast cancer detection using inception v3, inception v4 and modified inception mv4. *Neural Computing and Applications* 34, 333–348. doi: 10.1007/s00521-021-06372-1.
- Alzubaidi, L., Zhang, J., Humaidi, A.J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M.A., Al-Amidie, M., Farhan, L., 2021. Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions. *Journal of Big Data* 8, 1–74. doi: 10.1186/s40537-021-00444-8.
- Ariyarathne, G., De Silva, S., Dayarathna, S., Meedeniya, D., Jayarathne, S., 2020. ADHD identification using convolutional neural network with seed-based approach for fMRI data, in: *Proceedings of 9th International Conference on Software and Computer Applications (ICSCA)*, pp. 31–35. doi: 10.1145/3384544.3384552.
8. Bandara, M., Jayasundara, R., Ariyarathne, I., Meedeniya, D., Perera, C., 2023. Forest sound classification dataset: FSC22, *Sensors*, 23, 4:2032, doi: 10.3390/s23042032.
- Belousov, B., Abdulsamad, H., Klink, P., Parisi, S., Peters, J., 2021. Reinforcement learning algorithms: Analysis and applications. Springer.
10. Bozinovski, S., Fulgosí, A., 1976. The influence of pattern similarity and transfer learning upon training of a base perceptron b2, in: *Proc. Symposium Informatica*, pp. 121–126. doi: 10.31449/inf.v44i3.2828.
11. Brendan McMahan, H., Moore, E., Ramage, D., Hampson, S., Agüera y Arcas, B., 2016. Communication-efficient learning of deep networks from decentralized data. *arXiv-prints*, arXiv–1602 doi: 10.48550/arXiv.1602.05629.
12. Brownlee, J., 2018. Better deep learning: train faster, reduce overfitting, and make better predictions. *Machine Learning Mastery*.
13. Chauhan, N.K., Singh, K., 2018. A review on conventional machine learning vs deep learning, in: *Proc. International conference on computing, power and communication technologies (GUCON)*, IEEE. pp. 347–352. doi: 10.1109/gucon.2018.8675097.
14. Chitty-Venkata, K.T., Somani, A.K., 2022. Neural architecture search survey: A hardware perspective. *ACM Computing Surveys (CSUR)*, 55(4):78, PP. 1-36. doi: 10.1145/3524500.
- Chollet, F., 2017. Xception: Deep learning with depth-wise separable convolutions, in: *Proc. International Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1800–1807. doi: 10.1109/CVPR.2017.195.
- Dasanayaka, S., Shantha, V., Silva, S., Ambegoda, T., Meedeniya, D., 2022a. Interpretable machine learning for brain tumor analysis using MRI, in: *Proceedings of the 2nd International Conference on Advanced Research in Computing (ICARC)*, pp. 212–217. doi: 10.1109/ICARC54489.2022.9754131.
- Dasanayaka, S., Shantha, V., Silva, S., Meedeniya, D., Ambegoda, T., 2022b. Interpretable machine learning for brain tumour analysis using MRI and whole slide images. *Software Impacts* 13, 100340. doi: 10.1016/j.simpa.2022.100340.
- De Silva, S., Dayarathna, S., Ariyarathne, G., Meedeniya, D., Jayarathna, S., Michalek, A.M., 2021. Computational decision support system for ADHD identification. *International Journal of Automation and Computing (IJAC)* 18, 233–255. doi: 10.1007/s11633-020-1252-1.
- De Silva, S., Dayarathna, S., Meedeniya, D., 2022. Alzheimer's disease diagnosis using functional and structural neuroimaging modalities, in: Wadhera, T. and Kakkar, D. (Ed.),

- Enabling technology for neurodevelopmental disorders from diagnosis to rehabilitation. Taylor and Francis CRS Press, Routledge. chapter 11, pp. 162–183. doi: 10.4324/9781003165569-11.
- De Silva, S., Dayarathna, S.U., Ariyaratne, G., Meedeniya, D., Jayarathna, S., 2021b. fMRI feature extraction model for ADHD classification using convolutional neural network. International Journal of E-Health and Medical Communications (IJEHMC) 12, 81–105. doi:10.4018/IJEHMC.2021010106.
- Demotte, P., Wijegunaratna, K., Meedeniya, D., Perera, I., 2021. Enhanced sentiment extraction architecture for social media content analysis using capsule networks. Multimedia Tools and Applications doi: 10.1007/s11042-021-11471-1.
- Desai, M., Shah, M., 2021. An anatomization on breast cancer detection and diagnosis employing multi-layer perceptron neural network (MLP) and convolutional neural network (CNN). Clinical eHealth 4, 1–11. doi: 10.1016/j.ceh.2020.11.002.
- Dong, S., Wang, P., Abbas, K., 2021. A survey on deep learning and its applications. Computer Science Review 40, 100379. doi: 10.1016/j.cosrev.2021.100379.
- Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al., 2021. An image is worth 16x16 words: Transformers for image recognition at scale. in: Proceedings of the The International Conference on Learning Representations (ICLR) , pp. 1–22.
- Eelbode, T., Sinonquel, P., Maes, F., Bisschops, R., 2021. Pitfalls in training and validation of deep learning systems. Best Practice & Research Clinical Gastroenterology 52, 101712. doi: 10.1016/j.bpg.2020.101712.
- Ekman, M., 2021. Learning deep learning: Theory and practice of neural networks, computer vision, NLP, and transformers using TensorFlow. Addison-Wesley Professional.
- Fernando, C., Kolonne, S., Kumarasinghe, H., Meedeniya, D., 2022. Chest radiographs classification using multi-model deep learning: A comparative study, in: Proceedings of the 2nd International Conference on Advanced Research in Computing (ICARC), pp. 165–170. doi: 10.1109/ICARC54489.2022.9753811.
- Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y., 2020. Generative adversarial nets, Communications of the ACM 63, 139–144, doi: 10.1145/342262.
- Óeron, A., 2018. Neural networks and deep learning. O'Reilly Media, Inc.
- He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778. doi: 10.1109/CVPR.2016.90.
- He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778. doi:10.1109/CVPR.2016.90.
- Herath, L., Meedeniya, D., Marasingha, J., Weerasinghe, V., 2021. Autism spectrum disorder diagnosis support model using inceptionv3, in: Proceedings of International Research Conference on Smart Computing and Systems Engineering (SCSE), pp. 1–7. doi: 10.1109/SCSE53661.2021.9568314.
- Herath, L., Meedeniya, D., Marasingha, J., Weerasinghe, V., 2022. Optimize transfer learning for autism spectrum disorder classification with neuroimaging: A comparative study, in: Proceedings of the 2nd International Conference on Advanced Research in Computing (ICARC), pp. 171–176. doi: 10.1109/ICARC54489.2022.9753949.
- Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H., 2017. Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint. doi: 10.48550/arXiv.1704.04861.
- Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q., 2017. Densely connected convolutional networks, in: Proc. International Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2261–2269. doi: 10.1109/CVPR.2017.243.
- Hutter, F., Kotthoff, L., Vanschoren, J., 2019. Automated machine learning: methods, systems, challenges. Springer Nature. doi: 10.1007/978-3-030-05318-5.
- Iandola, F.N., Han, S., Moskewicz, M.W., Ashraf, K., Dally, W.J., Keutzer, K., 2016. SqueezeNet: Alexnet-level accuracy with 50x fewer parameters and! 0.5 mb model size. arXiv preprint, 1–13. doi: 10.48550/arXiv.1602.07360.
- Joseph, A.D., Nelson, B., Rubinstein, B.I.P., Tygar, J.D., 2019. Adversarial machine learning. Cambridge University Press. doi: 10.1017/9781107338548.

- Kang, M., Ko, E., Mersha, T.B., 2022. A roadmap for multi-omics data integration using deep learning. *Briefings in Bioinformatics* 23, bbab454. doi: 10.1093/bib/bbab454.
- Kapadnis, S., Tiwari, N., Chawla, M., 2022. Developments in capsule network architecture: A review. *Intelligent Data Engineering and Analytics* 266, 81–90. doi: 10.1007/978-981-16-6624-7_9.
- Ketkar, N., Santana, E., 2017. Deep learning with Python. volume 1. Springer. doi: 10.1007/978-1-4842-2766-4.
- Kumar, S., Kaur, P., Gosain, A., 2022. A comprehensive survey on ensemble methods, in: Proc. International conference for Convergence in Technology (I2CT), pp. 1–7, doi: 10.1109/I2CT54291.2022.9825269.
- Kumarasinghe, H., Kolonne, S., Fernando, C., Meedeniya, D., 2022. U-net based chest x-ray segmentation with ensemble classification for COVID-19 and pneumonia. *International Journal of Online and Biomedical Engineering (IJOE)* 18, 161–174. doi: 10.3991/joe.v18i07.30807.
- Ladosz, P., Weng, L., Kim, M., Oh, H., 2022. Exploration in deep reinforcement learning: A survey. *Information Fusion* 85, 1–22. doi: 10.1016/j.inffus.2022.03.003.
- Laxmisagar, H., Hanumantharaju, M., 2022. Detection of breast cancer with lightweight deep neural networks for histology image classification. *Critical Reviews™ in Biomedical Engineering* 50, 1–19. doi: 10.1615/CritRevBiomedEng.2022043417.
- Liu, X., Faes, L., Kale, A.U., Wagner, S.K., Fu, D.J., Bruynseels, A., Mahendiran, T., Moraes, G., Shamdas, M., Kern, C., et al., 2019. A comparison of deep learning performance against health-care professionals in detecting diseases from medical imaging: a systematic review and meta-analysis. *The Lancet Digital Health* 1, e271–e297. doi: 10.1016/s2589-7500(19)30123-2.
- Liu, Y., Sun, P., Wergeles, N., Shang, Y., 2021. A survey and performance evaluation of deep learning methods for small object detection. *Expert Systems with Applications* 172, 114602. doi: 10.1016/j.eswa.2021.114602.
- Ludwig, H., Baracaldo, N., 2022. Federated learning: A comprehensive overview of methods and applications. Springer Cham. doi: 10.1007/978-3-030-96896-0.
- Mahakalanda, I., Demotte, P., Perera, I., Meedeniya, D., Wijesuriya, W., Rodrigo, L., 2022. Chapter 7—deep learning-based prediction for stand age and land utilization of rubber plantation, in: Khan, M.A., Khan, R., Ansari, M.A. (Eds.), *Application of Machine Learning in Agriculture*. Elsevier Academic Press, pp. 131–156. doi: 10.1016/B978-0-323-90550-3.00008-4.
- Mandal, M., Vipparthi, S.K., 2021. An empirical review of deep learning frameworks for change detection: Model design, experimental frameworks, challenges and research needs. *IEEE Transactions on Intelligent Transportation Systems* 23, 6101–6122. doi: 10.1109/tits.2021.3077883.
- Mathew, A., Amudha, P., Sivakumari, S., 2020. Deep learning techniques: an overview, in: Proc. International conference on advanced machine learning technologies and applications, pp. 599–608. doi: 10.1007/978-981-15-3383-954.
- Meedeniya, D., Kumarasinghe, H., Kolonne, S., Fernando, C., De la Torre Díez, I., Marques, G., 2022a. Chest x-ray analysis empowered with deep learning: A systematic review. *Applied Soft Computing*, 109319. doi: 10.1016/j.asoc.2022.109319.
- Meedeniya, D., Mahakalanda, I., Lenadora, D., Perera, I., Hewawalpita, S., Abeysinghe, C., Nayak, S., 2022b. Chapter 13—Prediction of paddy cultivation using deep learning on land cover variation for sustainable agriculture, in: Poonia, R.C., Singh, V., Nayak, S.R. (Eds.), *Deep learning for sustainable agriculture*. Elsevier Academic Press. pp. 325–355. doi: 10.1016/B978-0-323-85214-2.00009-4.
- Meedeniya, D., Rubasinghe, I., 2020. A review of supportive computational approaches for neurological disorder identification, in: Wadhera, T., Kakkar, D. (Eds.), *Interdisciplinary approaches to altering neurodevelopmental disorder*. IGI Global. chapter 16, pp. 271–302. doi: 10.4018/978-1-7998-3069-6.ch016.
- Nagrath, P., Jain, R., Madan, A., Arora, R., Kataria, P., Hemanth, J., 2021. Ssdmvn2: A real time DNN-based face mask detection system using single shot multibox detector and mobilenetv2. *Sustainable Cities and Society* 66, 102692. doi: 10.1016/j.scs.2020.102692.
- Nguyen, D.C., Pham, Q.V., Pathirana, P.N., Ding, M., Seneviratne, A., Lin, Z., Dobre, O., Hwang, W.J., 2022. Federated learning for smart healthcare: A survey. *ACM Computing Surveys (CSUR)* 55, 1–37. doi: 10.1145/3501296.
- Nielsen, M.A., 2015. Neural networks and deep learning. Volume 25. Determination press San Francisco, USA.

- Opitz, D., Maclin, R., 1999. Popular ensemble methods: An empirical study. *Journal of Artificial Intelligence Research* 11, 169–198. doi: 10.1613/jair.614.
- Padmasiri, H., Madurawe, R., Abeysinghe, C., Meedeniya, D., 2020. Automated vehicle parking occupancy detection in real-time, in: Proceedings of 2020 Moratuwa Engineering Research Conference (MERCon), pp. 1–6. doi: 10.1109/MERCon50084.2020.9185199.
- Padmasiri, H., Shashirangana, J., Meedeniya, D., Rana, O., Perera, C., 2022. Automated license plate recognition for resource-constrained environments. *Sensors* 22.1434. doi: 10.3390/s22041434.
- Pathirana, P., Senarath, S., Meedeniya, D., Jayarathna, S., 2022a. Eye gaze estimation: A survey on deep learning-based approaches. *Expert Systems with Applications* 19, 1–16. doi: 10.1016/j.eswa.2022.116894.
- Pathirana, P., Senarath, S., Meedeniya, D., Jayarathna, S., 2022b. Single-user 2D gaze estimation in retail environment using deep learning, in: Proc. of the 2nd International Conference on Advanced Research in Computing (ICARC), pp. 206–211. doi: 10.1109/ICARC54489.2022.9754167.
- Qian, C., Zhu, J., Shen, Y., Jiang, Q., Zhang, Q., 2022. Deep transfer learning in mechanical intelligent fault diagnosis: Application and challenge. *Neural Processing Letters* 54, 2509–2531. doi: 10.1007/s11063-021-10719-z.
- Ravichandiran, S., 2019a. Hands-on deep learning algorithms with Python: Master deep learning algorithms with extensive math by implementing them using TensorFlow. Packt Publishing Ltd.
- Romo-Montiel, E., Menchaca-Mendez, R., Rivero-Angeles, M.E., Menchaca-Mendez, R., 2022. Improving communication protocols in smart cities with transformers. *ICT Express* 1, 50–55. doi: 10.1016/j.icte.2022.02.006.
- Ronneberger, O., Fischer, P., Brox, T., 2015. U-net: Convolutional networks for biomedical image segmentation, in: Proc. International Conference on Medical image computing and computer-assisted intervention, pp. 234–241. doi: 10.1007/978-3-319-24574-4-28.
- Rubasinghe, I., Meedeniya, D., 2019. Ultrasound nerve segmentation using deep probabilistic programming. *Journal of ICT Research and Applications* 13, 241–256. doi: 10.5614/itbj.ict.res.appl.2019.13.3.5.
- Rubasinghe, I., Meedeniya, D., 2020. Automated neuroscience decision support framework, in: Agarwal, B., Balas, V., Jain, L., Poonia, R., Manisha (Eds.), Deep learning techniques for biomedical and health informatics. Elsevier. chapter 13, pp. 305–326. doi: 10.1016/B978-0-12-819061-6.00013-6.
- Sabour, S., Frosst, N., Hinton, G.E., 2017. Dynamic routing between capsules, in: Proc. International Conference on Neural Information Processing Systems (NIPS), pp. 3859–3869. doi: 10.48550/arXiv.1710.09829.
- Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L., 2018. Mobilenetv2: Inverted residuals and linear bottlenecks, in: Proc. International Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4510–4520. doi: 10.1109/CVPR.2018.00474.
- Senarath, S., Pathirana, P., Meedeniya, D., Jayarathna, S., 2022a. Customer gaze estimation in retail using deep learning. *IEEE Access* 10, 64904–64919. doi: 10.1109/ACCESS.2022.3183357.
- Senarath, S., Pathirana, P., Meedeniya, D., Jayarathna, S., 2022b. Retail gaze: A dataset for gaze estimation in retail environments, in: Proceedings of the 3rd International Conference on Decision Aid Sciences and Applications (DASA), pp. 1040–1044. doi: 10.1109/DASA54658.2022.9765224.
- Sewak, M., Karim, M.R., Pujari, P., 2018. Practical convolutional neural networks: Implement advanced deep learning models using Python. Packt Publishing Ltd.
- Shafiq, M., Gu, Z., 2022. Deep residual learning for image recognition: A survey. *Applied Sciences* 12, 8972. doi: 10.3390/app12188972.
- Shashirangana, J., Padmasiri, H., Meedeniya, D., Perera, C., 2021a. Automated license plate recognition: A survey on methods and techniques. *IEEE Access* 9, 11203–11225. doi: 10.1109/ACCESS.2020.3047929.
- Shashirangana, J., Padmasiri, H., Meedeniya, D., Perera, C., Nayak, S.R., Nayak, J., Vimal, S., Kadry, S., 2021b. License plate recognition using neural architecture search for edge devices. *International Journal of Intelligent Systems (IJIS)* 36, 1–38. doi: 10.1002/int.22471.
- Shrestha, A., Mahmood, A., 2019. Review of deep learning algorithms and architectures. *IEEE Access* 7, 53040–53065. doi: 10.1109/access.2019.2912200.

- Shyamalee, T., Meedeniya, D., 2022a. Attention u-net for glaucoma identification using fundus image segmentation, in: Proceedings of the 3rd International Conference on Decision Aid Sciences and Applications (DASA), pp. 6–10. doi: 10.1109/DASA54658.2022.9765303.
- Shyamalee, T., Meedeniya, D., 2022b. CNN based fundus images classification for glaucoma identification, in: Proceedings of the 2nd International Conference on Advanced Research in Computing (ICARC), pp. 200–205. doi: 10.1109/ICARC54489.2022.9754171.
- Simonyan, K., Zisserman, A., 2014. Very deep convolutional networks for largescale image recognition. arXiv preprint arXiv:1409.1556, arXiv:1409.1556.
- Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A., 2015. Going deeper with convolutions, in: Proc. International Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1–9. doi: 10.1109/cvpr.2015.7298594.
- Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z., 2016. Rethinking the inception architecture for computer vision, in: Proc. International Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2818–2826. doi: 10.1109/cvpr.2016.308.
- Tan, M., Le, Q., 2019. Efficientnet: Rethinking model scaling for convolutional neural networks, in: Proc. International Conference on Machine Learning, pp. 6105–6114. doi: 10.48550/arXiv.1905.11946.
- Thomas, J.J., Karagoz, P., Ahamed, B.B., Vasant, P., 2019. Deep learning techniques and optimization strategies in big data analytics. IGI Global. doi: 10.4018/978-1-7998-1192-3.
- Ugail, H., 2022. Deep learning in visual computing: Explanations and examples. CRC Press.
- Vasudevan, S.K., Pulari, S.R., Vasudevan, S., 2022. Deep learning: A comprehensive guide. Chapman and Hall/CRC.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I., 2017. Attention is all you need. Advances in Neural Information Processing Systems 30, 1–15. doi: 10.48550/arXiv.1706.03762.
- Wang, H.n., Liu, N., Zhang, Y.y., Feng, D.w., Huang, F., Li, D.s., Zhang, Y.m., 2020. Deep reinforcement learning: A survey. Frontiers of Information Technology & Electronic Engineering 21, 1726–1744. doi: 10.1631/FITEE.1900533.
- Wijethilake, N., Meedeniya, D., Chitraranjan, C., Perera, I., 2020. Survival prediction and risk estimation of glioma patients using mRNA expressions, in: Proceedings of 20th International Conference on Bioinformatics and Bioengineering (BIBE), pp. 35–42. doi: 10.1109/BIBE50027.2020.00014.
- Wijethilake, N., Meedeniya, D., Chitraranjan, C., Perera, I., Islam, M., Ren, H., 2021. Glioma survival analysis empowered with data engineering—a survey. IEEE Access 9, 43168–43191. doi: 10.1109/ACCESS.2021.3065965.
- Yan, W., 2021. Computational methods for deep learning. Springer.
- Yang, Q., Zhang, Y., Dai, W., Pan, S.J., 2020. Transfer learning. Cambridge University Press. doi: 10.1017/9781139061773.
- Yao, X., Wang, X., Karaca, Y., Xie, J., Wang, S., 2020. Glomerulus classification via an improved googlenet. IEEE Access 8, 176916–176923. doi: 10.1109/access.2020.3026567.
- You, A., Kim, J.K., Ryu, I.H., Yoo, T.K., 2022. Application of generative adversarial networks (GAN) for ophthalmology image domains: A survey. Eye and Vision 9, 1–19. doi: 10.1186/s40662-022-00277-3.
- Zhang, C., Ma, Y., 2012. Ensemble machine learning: methods and applications. Springer. doi: 10.1007/978-1-4419-9326-7.
- Zhang, J., Li, C., Yin, Y., Zhang, J., Grzegorzek, M., 2022a. Applications of artificial neural networks in microorganism image analysis: a comprehensive review from conventional multilayer perceptron to popular convolutional neural network and potential visual transformer. Artificial Intelligence Review 55, 1–58. doi: 10.1007/s10462-022-10192-7.
- Zhang, T., Gao, L., He, C., Zhang, M., Krishnamachari, B., Avestimehr, A.S., 2022b. Federated learning for the internet of things: Applications, challenges, and opportunities. IEEE Internet of Things Magazine 5, 24–29. doi: 10.1109/iotm.004.2100182.
- Zhao, W., Alwidian, S., Mahmoud, O.H., 2022a. Adversarial training methods for deep learning: A systematic review. Algorithms 15, 283. doi: 10.3390/a15080283.
- Zhou, T., Ye, X., Lu, H., Zheng, X., Qiu, S., Liu, Y., 2022. Dense convolutional network and its application in medical image analysis. BioMed Research International 2022, 2384830. doi: 10.1155/2022/2384830.

Zoph, B ., Le, Q.V. , 2016. Neural architecture search with reinforcement learning. arXiv preprint arXiv:1611.01578, 1—16. doi: 10.48550/arXiv.1611.01578.

Zouch, W ., Sagga, D ., Echtioui, A ., Khemakhem, R ., Ghorbel, M ., Mhiri, C ., Hamida, A.B. , 2022. Detection of covid-19 from CT and chest X-ray images using deep learning models. Annals of Biomedical Engineering 50, 825–835. doi: 10.1007/s10439-022-02958-5.