SAMPUL

Advances in Intelligent Systems and Computing 1038

Yaxin Bi
Rahul Bhatia
Supriya Kapoor *Editors*

Intelligent Systems and Applications

Proceedings of the 2019 Intelligent Systems Conference (IntelliSys) Volume 2

Springer

DAFTAR ISI

Contents

PANITIA PELAKSANA/PANITIA PENGARAH

Conference Committee

CHAIRS

Sebastiano Battiato - Conference Chair

Università di Catania, Italy

Sebastiano Battiato is a full professor of Computer Science at the University of Catania. He received his degree in Computer Science (summa cum laude) in 1995 from the University of Catania and his Ph.D. in Computer Science and Applied Mathematics from the University of Naples in 1999. From 1999 to 2003 he was the leader of the “Imaging” team at STMicroelectronics in Catania. He joined the Department of Mathematics and Computer Science at the University of Catania in 2004 (respectively as assistant professor, associate professor in 2011 and full professor in 2016). He has been Chairman of the Undergraduate Program in Computer Science (2012-2017), and Rector's delegate for Education: postgraduates and Phd (2013-2016). He is currently the Scientific Coordinator of the PhD Program in Computer Science at the University of Catania. He is involved in research and directorship of the IPLab research lab (http://iplab.dmi.unict.it). He coordinates IPLab’s participation on large scale projects funded by national and international funding bodies, as well as by private companies. Prof. Battiato has participated as principal investigator in many international and national research projects. His research interests include Computer Vision, Imaging technology and Multimedia Forensics. He has edited 6 books and co-authored about 200 papers in international journals, conference proceedings and book chapters and has also been a guest editor of several special issues published in International Journals. He is also co-inventor of 22 international patents, reviewer for several international journals, and has been a regular member of numerous international conference committees.

Kohei Arai - Program Chair

Saga University, Japan

Dr Kohei Arai, a Scientist, Professor and Author. He is currently Professor at Saga University, Japan and Adjunct Prof. of the University of Arizona, USA since 1998. Dr Arai received PhD degree in Information Science from Nihon University in June 1982 and MS degree in Electronics Engineering from Nihon University in March 1974. His current research concerns are Satellite Remote Sensing, Radiative Transfer Equation, Human-Computer Interaction, Image Recognition and Understanding, Non-Linear Optimization Theory and Wavelet Analysis. Dr Arai holds 42 patents and received numerous awards, including the Patent Award of the Year. Dr Arai has been featured in Japan Times and Italian Newspapers for his work on Eyes only Computer System. He has worked on several global research collaboration projects during his career. He wrote 31 books and published 490 journal papers and 390 of conference papers.

ADVISORY COMMITEE

Ann Cavoukian

Distinguished Expert-in-Residence, Privacy by Design Centre of Excellence (Ryerson University), Canada

Dr. Ann Cavoukian is recognized as one of the world's leading privacy experts. Dr. Cavoukian served an unprecedented three terms as the Information & Privacy Commissioner of Ontario, Canada. There she created Privacy by Design, a framework that seeks to proactively embed privacy into the design specifications of information technologies, networked infrastructure and business practices, thereby achieving the strongest protection possible. In 2010, International Privacy Regulators unanimously passed a Resolution recognizing Privacy by Design as an International Standard. Since then, PbD has been translated into 40 languages! In 2018, PbD was included in a sweeping new law in the EU: the General Data Protection Regulation. Dr. Cavoukian is now the Executive Director of the Global Privacy & Security by Design Centre. She is also a Senior Fellow of the Ted Rogers Leadership Centre at Ryerson University, and a Faculty Fellow of the Center for Law, Science & Innovation at the Sandra Day O'Connor College of Law at Arizona State University. Dr. Cavoukian is the author of two books, "The Privacy Payoff: How Successful Businesses Build Customer Trust" with Tyler Hamilton, and "Who Knows: Safeguarding Your Privacy in a Networked World" with Don Tapscott. She has received numerous awards recognizing her leadership in privacy, including being named as one of the Top 25 Women of Influence in Canada, named as one of the Top 10 Women in Data Security and Privacy, and named as one of the 'Power 50' by Canadian Business. She was awarded the Meritorious Service Medal by the Governor General of Canada for her outstanding work on creating Privacy by Design and taking it global (May, 2017), named as one of the 50 Most Impactful Smart Cities Leaders (November, 2017), named among the Top Women in Tech (December, 2017), was awarded the Toastmasters Communication and Leadership Award (April, 2018), recognized among the Top 100 Identity Influencers (February, 2019), and most recently, she was named among the Top 18 Global AI Influencers within the AI & Tech Space (February, 2019).

Kohei Arai

Saga University, Japan

Dr Kohei Arai, a Scientist, Professor and Author. He is currently Professor at Saga University, Japan and Adjunct Prof. of the University of Arizona, USA since 1998. Dr Arai received PhD degree in Information Science from Nihon University in June 1982 and MS degree in Electronics Engineering from Nihon University in March 1974. His current research concerns are Satellite Remote Sensing, Radiative Transfer Equation, Human-Computer Interaction, Image Recognition and Understanding, Non-Linear Optimization Theory and Wavelet Analysis. Dr Arai holds 42 patents and received numerous awards, including the Patent Award of the Year. Dr Arai has been featured in Japan Times and Italian Newspapers for his work on Eyes only Computer System. He has worked on several global research collaboration projects during his career. He wrote 31 books and published 490 journal papers and 390 of conference papers.

Nikola Serbedzija

ITETA Consulting, Germany, Germany

Dr. Nikola Šerbedžija is the founder of iTETA Consulting, specialized in deploying advanced information technology achievements in praxis, offering support in digital transformation, research on future technologies, coaching and teaching. He was scientific advisor at Fraunhofer FOKUS [1992-2018], responsible for new research initiatives and innovative technology, visiting professor at University of Sydney [1999-2000] and at the University of Arts, Berlin [2000-2007]. His research work is primarily concerned with distributed computing, especially applied to ubiquitous systems, cyber-physical systems and IoT. In addition, his research covers modelling, architectural approach and practical deployment of adaptive and autonomous systems. He was principal investigator in the various research projects that led to REFLECT and ASCENS frameworks for developing pervasive-adaptive systems and self-aware autonomous systems, respectively. He has been acting as an expert for European FP7 and Horizon 2020 programmes, mostly for the Future Emerging Technology actions. He has been on numerous international conference committees and held various invited/keynote talks at scientific events. He co-authored many chapters in special edition monographies or books and he is the author of over 100 journal and conference papers.

Pascal LORENZ

University of Haute Alsace, France

Pascal Lorenz received his M.Sc. (1990) and Ph.D. (1994) from the University of Nancy, France. Between 1990 and 1995 he was a research engineer at WorldFIP Europe and at Alcatel-Alsthom. He is a professor at the University of Haute-Alsace, France, since 1995. His research interests include QoS, wireless networks and high-speed networks. He is the author/co-author of 3 books, 3 patents and 200 international publications in refereed journals and conferences. He was Technical Editor of the IEEE Communications Magazine Editorial Board (2000-2006), Chair of Vertical Issues in Communication Systems Technical Committee Cluster (2008-2009), Chair of the Communications Systems Integration and Modeling Technical Committee (2003-2009) and Chair of the Communications Software Technical Committee (2008-2010). He has served as Co-Program Chair of IEEE WCNC 2012, ICC 2004 and ICC 2017, tutorial chair of VTC 2013 Spring and WCNC 2010, track chair of PIMRC 2012, symposium Co-Chair at Globecom 2011-2007, ICC 2010-2008 and ICC 2014. He has served as Co-Guest Editor for special issues of IEEE Communications Magazine, Networks Magazine, Wireless Communications Magazine, Telecommunications Systems and LNCS. He is senior member of the IEEE, IARIA fellow and member of many international program committees. He has organized many conferences, chaired several technical sessions and gave tutorials at major international conferences.

Wenbing Zhao

Cleveland State University, United States

Dr. Wenbing Zhao is a Professor of Electrical and Computer Engineering at Cleveland State University, Cleveland, Ohio, USA. Dr. Zhao has done research in a number of different areas, including fault tolerance computing, computer and network security, smart and connected healthcare, machine learning and control, quantum optics and superconducting physics. Currently, his research focuses on dependable distributed computing, and smart and connected healthcare. He is also exploring an exciting emerging research area on medical cyber physical systems, where he could integrate dependable computing with smart and connected healthcare. Dr. Zhao's recent research has been funded by the National Science Foundation (CNS-0821319), US Department of Transportation (via CSU Transportation Center), and Cleveland State University (several FRD grants and a Faculty Scholarship Initiative grant). Dr. Zhao has authored a research monograph, titled "Building Dependable Distributed Systems," and contributed 22 book chapters and over 100 peer-reviewed journal and conference publications. He also has a US patent on consistent time service for fault tolerant distributed systems and another patent pending on privacy-aware human motion tracking. Dr. Zhao has been very active in professional services. Other than the Departmental/College/University duties, he has served on several NSF panels, and on the organizing and technical program committees of numerous conferences/workshops. Dr. Zhao is a senior member of Institute of Electrical and Electronics Engineers (IEEE). He is currently serving as a member of the executive board of the IEEE Cleveland Chapter.

Liming Chen

Ulster University, United Kingdom

Dr Chen is Professor of Data Analytics, Research Director for School of Computing at Ulster University, UK. His current research interests include data analytics, pervasive computing, artificial intelligence, user-centred intelligent systems and their applications in smart healthcare. Dr Chen is an IET Fellow, an IEEE Senior Member, a co-founder and co-director of the IEEE CIS" User-centred Smart Systems" Task Force. He is currently the co-investigator of the British Telecommunication Ireland Innovation Centre, and was the coordinator of the EU Horizon2020 MSCA ITN ACROSSING project and has serves as the principal investigator for the EU AAL PIA project, the MobileSage project and FP7 MICHELANGELO project, and a number of projects funded by industry and third countries. Dr Chen has over 230 publications in internationally recognised journals, book series and conferences. He is the general chair or program chair for IEEE Smart World Congress 2019, IEEE UIC2017, IEEE HealthCom2017, SAI Computing 2017, IEEE UIC2016, IntelliSys2016, MoMM2015/2014, SAI2015, IWAAL2014, UCAMI2013, and an organising chair of many workshops. He is an associate editor of IEEE THMS, and guest editors for IEEE Computer and THMS, Elsevier PMC and IJDSN and Springer PUC and AIHC. He has delivered nearly 30 talks, keynotes and seminars in various forums, conferences, industry and academic events. Luke has served as an expert for research funding assessment for UKRI, EU Horizon2020, Canada, Chile, Netherlands and Denmark.

Peter Mueller

IBM Zurich Research Laboratory, Switzerland

Peter Mueller joined IBM Research as a Research Staff Member in 1988. His research expertise covers broad areas of distributed computing systems architecture, microwave technology, device physics, nano science and modeling. His current field of research is in the areas of quantum technology and data center storage security. Peter is a founding member and was the Chair of the IEEE ComSoc Communications and Information Systems Security Technical Committee (CIS-TC).

David Stupples

City University London, United Kingdom

Professor David Stupples specialises in research and development of networked electronic and radio systems. For a number of years he undertook research in military surveillance systems at the Royal Signal and Radar Establishment at Malvern in the UK, followed by satellite surveillance and systems research for the UK Government. He then spent three years developing secure communications and signals processing for surveillance satellites and air defence systems for Hughes Aircraft Corporation in the US. In his early career, Dr David Stupples was employed in radar and communications systems development by the Royal Air Force. Later, he was a senior partner with PA Consulting Group where he was responsible for the company's consultancy work on the design, build and operation large-scale surveillance networks for world-wide clients. His current research is in networked systems with a focus on multi-sensor surveillance using both active and passive sensors. This research work has intelligence, military and commercial uses with the latter applying to future air traffic management. The research employs a variety of systems modelling techniques particularly to assess the behaviour of networked systems under stressed conditions in order to identify and assess emergent properties for their impact on both safety and integrity. Professor David Stupples has been applying this research to internet security, particularly focused on cyber terrorism and organised cyber crime for both the UK government and commercial companies. This work has required the development of new systems analysis and modelling techniques to understand the behaviour of complex internet systems. However, he still maintains an active interest in radar systems research; particularly synthetic aperture systems.

Yaxin Bi

University of Ulster, United Kingdom

Yaxin Bi is a Reader in School of Computing and Mathematics at University of Ulster. His research interests are multiple classification systems and ensemble learning, the Dempster-Shafer theory of evidence, big data analytics for satellite data exploitation, data/text mining for sentiment and media content analyses, and sensor fusion for understanding human behaviours in applications of Smart Homes. He has been involved in a number of European Frameworks projects since 1997 and has more than 100 peer-reviewed publications in international journals and conferences, including journals of Artificial Intelligence, IEEE Transactions of Knowledge and Data Engineering and International Journal of Approximate Reasoning. He was a program co-chair of the 4th International Conference on Knowledge Science, Engineering and management (KSEM) 2010 and a steering committee member of KSEM and SAI.

Chaomin Luo

Mississippi State University, United States

Chaomin Luo (Senior Member, IEEE) received the Ph.D. degree in electrical and computer engineering from the University of Waterloo, Waterloo, ON, Canada, the M.Sc. degree in engineering systems and computing from the University of Guelph, Guelph, ON, Canada, and the B.Eng. degree in electrical engineering from Southeast University, Nanjing, China. His research interests include robotics and autonomous systems, machine learning, and intelligent systems. Dr. Luo was the recipient of the Best Paper Awards in IEEE International Conference on Information and Automation, International Conference on Swarm Intelligence, and SWORD Conference. He is currently an Associate Editor for the International Journal of Robotics and Automation, International Journal of Swarm Intelligence Research and IEEE Transactions on Cognitive and Developmental Systems. He was an Associate Editor for 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems, and a Tutorials Co-Chair in 2020 IEEE Symposium Series on Computational Intelligence (IEEE-SSCI). He is currently an Associate Professor with the Department of Electrical and Computer Engineering, Mississippi State University, Mississippi State, MS, USA. Dr. Luo is active nationally and internationally in his research field. He was the panelist in 2017 NSF GRFP Panelist program. He was the Program Co-Chair in 2018 IEEE International Conference on Information and Automation (IEEE-ICIA'2018). He was the Plenary Session Co-Chair in the 2019 and 2018 International Conference on Swarm Intelligence, and he was the Invited Session Co-Chair in the 2017 International Conference on Swarm Intelligence. He was the General Co-Chair of the 1st IEEE International Workshop on Computational Intelligence in Smart Technologies (IEEE-CIST 2015), and Journal Special Issues Chair, IEEE 2016 International Conference on Smart Technologies (IEEE-SmarTech), Cleveland, OH, USA. Also, he was Chair and Vice Chair of IEEE SEM - Computational Intelligence Chapter and was a Chair of IEEE SEM - Computational Intelligence Chapter and Chair of Education Committee of IEEE SEM.

TECHNICAL PROGRAM COMMITEE

Aashish Jain, *Salesforce*

Domain of Research: Multi-model Deep learning model, Self supervised Learning

Abolfazl Razi, *Northern Arizona University*

Domain of Research: Artificial intelligence, Wireless networking, Autonomous System Design

Abolfazl Gharaei, *PhD in Industrial Engineering, Department of Mechanical and Industrial Engineering, University of Toronto, Ontario, Canada*

Domain of Research:

Balakrishnan Subramanian, *Professor and Head, Department of Computer Science and Business Systems, Sri Krishna College of Engineering and Technology*

Domain of Research: Artificial Intelligence, Cloud Computing, IoT

Bálint Molnár, *ELTE, Eötvös Loránd University, Budapest*

Domain of Research: Information Systems

Benjamin Guedj, *Inria and University College London and The Alan Turing Institute*

Domain of Research: Machine learning

Bhavtosh Rath, *TARGET CORPORATION*

Domain of Research: Recommender systems, Social Network Analysis

Biju Issac, *Northumbria University, Newcastle*

Domain of Research: Intelligent Systems,Artificial Intelligence,Security,Cloud Computing,Computational Intelligence,Computer Networks,Cyber Security,Internet of Things,Network Security

João Paulo de Sousa, *Polytechnic Institute of Bragança (IPB)*

Domain of Research: Games, Artificial Intelligence, System Administration

João Pedro Carvalho, *Lusófona University, COPELABS*

Domain of Research: Machine Vision,Artificial Intelligence,Artificial Intelligence,Computer Vision,Data Science

JOHN MEYER, *utrecht university*

Domain of Research: AI, in particular agent technology

JONGMOO CHOI, *University of Southern California*

Domain of Research: Computer Vision, Robotics

Jordi Garcia, *UPC BarcelonaTECH*

Domain of Research: Smart cities technology, SC architectures, edge computing, data management in smart environments

Kostas Kolomvatsos, *Senior Researcher*

Domain of Research: Expert Systems

Krzysztof Hauke, *Wroclaw University of Economics*

Domain of Research: innovation, creativity, intelligent systems, new ICT technologies

Kun Chang Lee, *SKK Business School, Sungkyunkwan University*

Domain of Research: intelligent systems applied to decision making, neuro-physiological analysis of intelligent decision making

Larbi Boubchir, *University of Paris 8*

Domain of Research: Artificial Intelligence,Biometrics

leila halawi, *Embry Riddle Aeronautical University*

Domain of Research: BI, Analytics, IOT

Leszek Ziora, *Czestochowa University of Technology*

Domain of Research: Business analytics, Business Intelligence systems, big data, data mining, artificial intelligence, Management Information Systems , cloud computing, e-learning, IT security.

Luís Pinto Ferreira, *ISEP – School of Engineering, Polytechnic of Porto, Portugal*

Domain of Research: Industrial Simulation; Discrete Event simulation

M. Hassaballah, *Department of Computer Science, Faculty of Computers and Information,South Valley University, Qena, Egypt.*

Domain of Research: ,

M. Zakaria KURDI, *University of Lynchburg*

Domain of Research: NLP, Text Mining

Maciej Wielgosz, *AGH University of Science and Technology*

Domain of Research: artificial intelligence, cognitive computing, hardware acceleration

Mahdi Imani, *Texas A&M University*

Domain of Research: Machine Learning, Computational Biology and Bioinformatics

Ouyang Zhang, *Google LLC*

Domain of Research: Communication,Software Engineering and Quality,3G/4G Network Evolutions,Cloud Computing,Cognitive Radio,Communication,Human Computer Interaction,Intrusion Detection,Localization

Ozdemir Cetin, *Dr*

Domain of Research: Computational Pathology

Paolino Di Felice, *University of L'Aquila*

Domain of Research: Software engineering, Software development, Software reusability, Model-Driven Engineering, Spatial databases, digital maturity models

Pasquale De Luca, *Department of Science and Technology, University of Naples Parthenope*

Domain of Research: Intelligent Systems,Parallel and Distributed Computing,Parallel Computing

Rui Bian, *University of Delaware*

Domain of Research: ,

Rushdi Hamamreh, *Al-Quds University*

Domain of Research: Information security, Mobile Networks, Intelligent systems

Saad A. Aldoihi, *KACST*

Domain of Research: Technology Trends,Intelligent Systems,Artificial Intelligence,Artificial Intelligence,Assistive Robots,Big Data,Robotics

Saeid Nahavandi, *Deakin University*

Domain of Research: Intelligent Systems, Robotics and Haptics

Vojislav Stojkovic, *Morgan State University*

Domain of Research: Intelligent Agents

Vusumuzi Malele, *National Department of Science and Innovation*

Domain of Research: Systems engineering, engineering management, innovation, ICT, wireless and mobile networks/computing

Wadii Boulila, *Prince Sultan university*

Domain of Research: Machine Vision,Intelligent Systems,Artificial Intelligence,

WEI DAI DAI, *University of Arkansas at Little Rock*

Domain of Research: Machine learning, digital images, and data quality

Weiqi Yan, *Auckland U of Technology*

Domain of Research: Deep Learning, Computer Vision, Intelligent Surveillance

Weiwei Shen, *GE Global Research Center*

Domain of Research: computational finance, machine learning, simulation and numeric methods

Wen-Cheng Lai, *National Taiwan University of Science and Technology*

Domain of Research: Computer and Communication

CONFERENCE MANAGER

Supriya Kapoor

The Science and Information Organization

Supriya Kapoor oversees all conference management activities for the organization including but not limited public and corporate relations, messaging, on-going communications, media outreach and material development. Supriya is happy to work for an organization that changes the world through global interaction and wisdom sharing. She has a breadth of experience in prospecting, relations, training, presentations, publication and financial planning. Supriya has built extensive networks that bring professional growth to the conferences. Before directing herself to management role, Supriya was a subject matter expert in the area of database and backend technologies.

Advances in Intelligent Systems and Computing

Volume 1038

Series Editor

Janusz Kacprzyk, Systems Research Institute, Polish Academy of Sciences, Warsaw, Poland

Advisory Editors

Nikhil R. Pal, Indian Statistical Institute, Kolkata, India
Rafael Bello Perez, Faculty of Mathematics, Physics and Computing, Universidad Central de Las Villas, Santa Clara, Cuba
Emilio S. Corchado, University of Salamanca, Salamanca, Spain
Hani Hagras, School of Computer Science and Electronic Engineering, University of Essex, Colchester, UK
László T. Kóczy, Department of Automation, Széchenyi István University, Gyor, Hungary
Vladik Kreinovich, Department of Computer Science, University of Texas at El Paso, El Paso, TX, USA
Chin-Teng Lin, Department of Electrical Engineering, National Chiao Tung University, Hsinchu, Taiwan
Jie Lu, Faculty of Engineering and Information Technology, University of Technology Sydney, Sydney, NSW, Australia
Patricia Melin, Graduate Program of Computer Science, Tijuana Institute of Technology, Tijuana, Mexico
Nadia Nedjah, Department of Electronics Engineering, University of Rio de Janeiro, Rio de Janeiro, Brazil
Ngoc Thanh Nguyen, Faculty of Computer Science and Management, Wrocław University of Technology, Wrocław, Poland
Jun Wang, Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong, Shatin, Hong Kong

The series "Advances in Intelligent Systems and Computing" contains publications on theory, applications, and design methods of Intelligent Systems and Intelligent Computing. Virtually all disciplines such as engineering, natural sciences, computer and information science, ICT, economics, business, e-commerce, environment, healthcare, life science are covered. The list of topics spans all the areas of modern intelligent systems and computing such as: computational intelligence, soft computing including neural networks, fuzzy systems, evolutionary computing and the fusion of these paradigms, social intelligence, ambient intelligence, computational neuroscience, artificial life, virtual worlds and society, cognitive science and systems, Perception and Vision, DNA and immune based systems, self-organizing and adaptive systems, e-Learning and teaching, human-centered and human-centric computing, recommender systems, intelligent control, robotics and mechatronics including human-machine teaming, knowledge-based paradigms, learning paradigms, machine ethics, intelligent data analysis, knowledge management, intelligent agents, intelligent decision making and support, intelligent network security, trust management, interactive entertainment, Web intelligence and multimedia.

The publications within "Advances in Intelligent Systems and Computing" are primarily proceedings of important conferences, symposia and congresses. They cover significant recent developments in the field, both of a foundational and applicable character. An important characteristic feature of the series is the short publication time and world-wide distribution. This permits a rapid and broad dissemination of research results.

**** Indexing: The books of this series are submitted to ISI Proceedings, EI-Compendex, DBLP, SCOPUS, Google Scholar and Springerlink ****

More information about this series at http://www.springer.com/series/11156

Yaxin Bi · Rahul Bhatia · Supriya Kapoor
Editors

Intelligent Systems and Applications

Proceedings of the 2019 Intelligent Systems Conference (IntelliSys) Volume 2

Springer

Editors
Yaxin Bi
School of Computing, Computer Science Research Institute
Ulster University
Newtownabbey, UK

Rahul Bhatia
The Science and Information (SAI) Organization
Bradford, West Yorkshire, UK

Supriya Kapoor
The Science and Information (SAI) Organization
Bradford, West Yorkshire, UK

ISSN 2194-5357 ISSN 2194-5365 (electronic)
Advances in Intelligent Systems and Computing
ISBN 978-3-030-29512-7 ISBN 978-3-030-29513-4 (eBook)
https://doi.org/10.1007/978-3-030-29513-4

© Springer Nature Switzerland AG 2020
This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.
The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.
The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

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

Editor's Preface

The Intelligent Systems Conference (IntelliSys) 2019 was held on September 5 and 6, 2019, in London, UK. The Intelligent Systems Conference is a prestigious annual conference on areas of intelligent systems and artificial intelligence and their applications to the real world, which is built on the success of the IntelliSys conferences in the past five years held at London.

This conference not only presented the state-of-the-art methods and valuable experience from researchers in the related research areas, but also provided the audience with a vision of further development in the fields. The research that comes out of a series of the IntelliSys conferences will provide insights into the complex intelligent systems and pave a way for the future development.

The Program Committee of IntelliSys 2019 represented 25 countries, and the authors submitted 546 papers from 45 countries. This certainly attests to the widespread international importance of the theme of the conference. Each paper was reviewed on the basis of originality, novelty and rigorousness. After the reviews, 223 were accepted for presentation, out of which 189 papers are finally being published in the proceedings.

The event was a two-day program comprised of 24 paper presentation sessions and poster presentations. The themes of the contributions and scientific sessions ranged from theories to applications, reflecting a wide spectrum of artificial intelligence. We are very gratified to have an exciting lineup of featured speakers who are among the leaders in changing the landscape of artificial intelligence and its application areas. Plenary speakers include: Grega Milcinski (CEO, Sinergise), Detlef D Nauck (Chief Research Scientist for Data Science at BT Technology), Giulio Sandini (Director of Research - Italian Institute of Technology) and Iain Brown (Head of Data Science, SAS UK&I).

The conference would truly not function without the contributions and support received from authors, participants, keynote speakers, program committee members, session chairs, organizing committee members, steering committee members and others in their various roles. Their valuable support, suggestions, dedicated commitment and hard work have made the IntelliSys 2019 successful.

It has been a great honor to serve as the general chair for the IntelliSys 2019 and to work with the conference team. We believe this event will certainly help further disseminate new ideas and inspire more international collaborations.

Kind Regards,
Yaxin Bi
Conference Chair

ARTIKEL

Determining the Number of Hidden Layers in Neural Network by Using Principal Component Analysis

Muh. Ibnu Choldun R.(✉), Judhi Santoso, and Kridanto Surendro

School of Electrical Engineering and Informatics, Institut Teknologi Bandung, Jl. Ganecha 10, Bandung, Indonesia
ibnucholdun@poltekpos.ac.id, judhi@stei.itb.ac.id, endro@informatika.org

Abstract. One of the challenges faced in the success of Deep Neural Network (DNN) implementation is setting the values for various hyper-parameters, one of which is network topology that is closely related to the number of hidden layers and neurons. Determining the number of hidden layers and neurons is very important and influential in DNN learning performance. However, up to now, there has been no guidance on it. Determining these two numbers manually (usually through trial and error methods) to find fairly optimal arrangement is a time-consuming process. In this study, we propose the method used for determining the number of hidden layers was through the number of components formed on the principal component analysis (PCA). By using Forest Type Mapping Data Set, based on PCA analysis, it was found out that the number of hidden layers that provide the best accuracy was three. This is in accordance with the number of components formed in the principal component analysis which gave a cumulative variance of around 70%.

Keywords: Neural network · Hidden layer · PCA

1 Introduction

Neural network has been successfully applied in various fields such as computer science, finance, medicine, engineering, physics, and so forth. The main reason is that neural network could approach arbitrary function. Over the past 30 years, a number of researches showed that artificial neural network called as feedforward network with one hidden layer can approach all functions arbitrarily [1–6]. One of important aspects for designing neural network is architecture or its topology that is closely related to the capability of generalization [7, 8]. Since the era of deep learning, the use of neural network that has more than one hidden layer is a research topic that caught researchers' interest. One of the challenges in the success of deep neural network implementation is setting values for various hyper-parameters, one of which is network topology that is closely related to the number of hidden layers and neurons. Determining the number of hidden layers and neurons is very important and influential in the performance of deep neural network [7, 8]. Therefore research to determine both of them is very necessary.

© Springer Nature Switzerland AG 2020
Y. Bi et al. (Eds.): IntelliSys 2019, AISC 1038, pp. 490–500, 2020.
https://doi.org/10.1007/978-3-030-29513-4_36

According to some literatures aforementioned, determining the number of hidden layers or optimal number of neurons still has no clear guideline. Moreover, the roles and functions of both are explained minimally. Some literatures propose methods or ways to determine the number of hidden layers or neurons, but they are not generally applied due to the type of input and output data [9, 10]. Studies in this area still leave difficult research tasks [11]. Some of the methods mentioned are apparently not applicable for different types of data. The earlier researchers determined the number of hidden layers or neurons based on their experience. However, less experienced researchers even did it based on 'trial and error'. One of the methods that can be applied practically is through data mining techniques, clustering and regression [12–14]. In this study, the researcher did not explain the relationship between the number of clusters formed by feature extraction that is going to be used in machine learning process.

Although it is a difficult area of research, determining the number of hidden layers and neurons should be carried out. This is because they greatly determine the deep neural network learning performance. As a first step, a method or technique is needed to determine the number of hidden layers. Hidden layers in the neural network in stages represent increasingly complex features, while the main components of the PCA represent the information content that exists. By assuming that the complexity of features is in line with the size of the feature information, then in this study we propose a method specifically for determining the number of hidden layers in a neural network based on the number of main components formed on principal component analysis (PCA).

2 Related Work

Studies compared the use of one or two hidden layers focused on univariate and multivariate functions [4–6, 15]. Thomas [4, 5] got different result that the use of two hidden layers applied to predictive functions showed better performance. Guliyev and Ismailov [6] concluded that the use of one hidden layer was less capable of approaching the multivariate function, so the use of two hidden layers showed better performance.

To determine the number of hidden layers and neurons at once, the method is divided into two: automatically and manually [15]. Automatic determination uses models, but it also can be done without models. Manual determination uses certain formulas, some are done by trial and error. Automatic determination without a model consists of grid search (GS) and random search (RS). GS is generally used to optimize DNN parameters only if the number is very low. First, users choose a range of values to explore. Then, DNN is set for each specification along with other parameters. Usually, GS is carried out for logarithmic scale steps that the best combination is estimated progressively to search [16]. The obvious disadvantage of GS is its time complexity – parameter k takes different values (by assuming that n is the same for all parameters), the complexity grows exponentially at level $O(n^k)$. Random search (RS) is an alternative to GS. It integrates faster into parametricization. RS is not adaptive that it is not dynamically updated during the experiment (the solutions found did not do anything to

the search). Furthermore, there is also hybrid algorithm that combines RS with other techniques to improve its performance, for instance, manual updates provided by experts [16, 17]. The technique that has been done turns out to be not generally accepted.

Model-based algorithm builds replacement models for hyper-parameter space. Then, hyper-parameters are set by using this model. Most of these techniques use Bayesian regression model that turns this problem into a trade-off between exploration (traversing unknown areas of space that classification of the performance is unknown), and exploitation (analyzing hyper-parameters that are likely to perform well and closed to "good" hyper-parameters that have been investigated). One of model-based algorithms is the use Particle Swarm Optimization (PSO) [3, 15]. PSO is an optimization technique done by continuously calculating potential solutions by using a quality reference. This algorithm optimizes problems by moving particles or prospective solutions for problems through certain functions for the position and velocity of the particles. The particles' movement is influenced by the best solution for the particles and the best solution in general that is obtained from other particles. This set of particles is called as a swarm that in the end it will move towards the best solution.

Although some researchers have proposed methods to calculate the number of hidden layers and neurons at once, but the methods were rarely used by other researchers, because they are difficult to apply or do not match the type of dataset used, therefore some researchers decided to use trial and error methods [8]. Therefore, a method or technique is needed to calculate the number of hidden layers or hidden neurons that can be generally accepted.

3 Neural Network

Architecture of back propagation neural network consists of several layers: input layer, hidden layer and output layer. Each neuron in each layer is connected to the next layer. The neurons in the input layer are connected to neurons in the hidden layer through synapses called as weight. In addition, hidden layers of neurons connected through synapses with neurons in the output layer. One of the neural network architectures is called as Multilayer Perceptron (MLP) [18]. MLP consists of: input layers where the neurons are fully connected to each neuron in the first hidden layer, each neuron in the hidden layer is connected to each neuron in the next hidden layer, and each neuron in the last hidden layer is fully connected with each neuron in the output layer. MLP, or often called as feedforward deep network, is a classic example of deep learning model.

Figure 1 is an example of a neural network with an MLP architecture consisting of input layers, two hidden layers, and an output layer. The closer the hidden layer to the output layer the better it identifies the complex features. For instance, MLP architecture that uses three hidden layers for face identification: the first hidden layer is used to identify the geometric shape that can distinguish the facial parts, the second hidden layer is used to identify eyes, nose, mouth, etc., and the third hidden layer is used to identify the face.

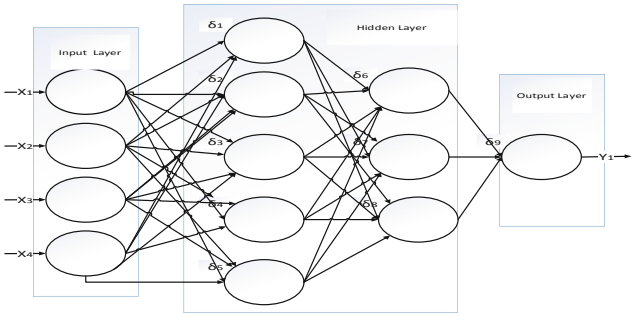


Fig. 1. Neural network

The training process for neural network is done by finding the value of each connection weight, which is done iteratively. Therefore, the output can correctly predict the target value of the training data examples. Backpropagation is an algorithm that is widely used for neural network learning, especially in MLP architecture. Each learning algorithm developed is based on inductive bias, a collection of assumptions that underlie the selection criteria of learning algorithm model. According to Mitchell [19], a learner's inductive bias is a set of additional assumptions that is sufficient to justify an inductive conclusion as a deductive conclusion. An example of an inductive bias that can be applied to MLP learning is to obtain the desired accuracy that the correct architecture must be chosen (the number of hidden layers and neurons for each hidden layer).

After obtaining the output from the perceptron based on the input given, at the error stage the perceptron calculation will be evaluated "whether the perceptron output matches the expected output or not". To calculate the difference between actual output and desired output can be used various functions of errors such as Squared Error (Eq. 1), Root Mean Squared Error (RMSE), or Cross Entropy (Eq. 2).

$$E = \frac{1}{2}\sum_{i=1}^{N}(t_i - y_i)^2 \tag{1}$$

for t = actual output, y = desired output, and N = number of data, then squared errors from t and y can be calculated using Eq. 1. The use of squared error often has problems if it is implemented in the case of classification with logistical output. Unlike the squared error, cross entropy does not experience problems in this case. In cross entropy, the greater the prediction error of the neural network, the greater the update value for the weight. The error cross entropy function can be formulated as follows:

$$E = -\frac{1}{N}\sum_{i=1}^{N} a_i \log(y_i) + (1 - a_i)\log(1 - y_i) \tag{2}$$

where y_i is the output of the neuron to i, a_i is the *actual output* to i, and N the total number of datasets.

4 Principal Component Analysis

Principal Component Analysis (PCA) is a multivariate analysis that transforms correlated origin variables into new variables that do not correlate with each other by reducing a number of the variables mentioned, so that the variables have smaller dimension but can explain most of the diversity of the origin variables [20]. The PCA steps are as follows.

1. There is a data matrix ($X = [x_1, x_2, \ldots, x_N]$), N is the total number of samples and xi represents the sample to i.
2. Calculate the average of all samples, as follows:

$$\mu = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{3}$$

3. Subtract all data samples with the average, as follows:

$$D = \{d_1, d_2, \ldots, d_N\} = \sum_{i=1}^{N} x_i - \mu \tag{4}$$

4. Calculate the covariance matrix, as follows:

$$\sum = \frac{1}{N-1} D \times D^T \tag{5}$$

5. Calculate the eigenvectors V and eigenvalues λ from the covariance matrix (Σ)
6. Sort the eigenvectors based on the eigenvalues that correspond
7. Select eigenvectors which have the largest eigenvalues $W = \{v_1, \ldots, v_k\}$. The selected Eigenvectors (W) represent the PCA projection space.
8. All samples are projected on the dimensional space lower than PCA (W), as follows:

$$Y = W^T D \tag{6}$$

The number of the main components formed is equal to the number of original variables. The selection of the main components used was based on the eigenvalue value that the main components will be used if the eigenvalue is greater than one. However, the reduction (simplification) of the dimension can also be done by finding the percentage criteria of the data diversity explained by the first few main components. For example, the number of the main components used was if the cumulative variance of the components was at least 70%.

5 Research Method

5.1 Research Framework

The more hidden layers in a neural network the more they can represent the increasingly complex features [19]. Deep architecture in neural network can detect features

arranged in each layer. The lower layers (that are closer to the input layer) could detect simpler features, while the layers closer to the ouput layer could detect more complex features [21]. Increasingly complex feature represents higher information content received. The high information in PCA is represented in the main component that has high variance. In other words, increasingly complex feature in hidden layers of the neural network is in accordance with the main components that have higher variance. As a result, the number of hidden layers in the required neural network will be in accordance with the number of the main components in the principal component analysis. Therefore, in this study, the number of hidden layers in the neural network could be determined based on the number of the main components formed through PCA.

5.2 Preparing the Dataset

The data set used in this study was Forest Type Mapping Data Set that was converted into numeric data. The data sets could be downloaded on UCI Machine Learning Repository website[1]. The number of input attribute was 27 and the output category was 4 classes: 's' ('Sugi' forest), 'h' ('Hinoki' forest), 'd' ('Mixed deciduous' forest), 'o' ('Other' non-forest land). The number of dataset used was 325 records, 70% of which was used as training data and the remaining is for testing. The output function used in the data set was classification.

5.3 Standardization and Normalization

Standardization was done by deleting data that had empty attributes and also eliminating extreme values. Normalization was done with the aim of getting data with attributes that are scaled accordingly. The normalization done was Min-Maks with a value between 0 and 1. Min-Maks normalization formula is as follows:

$$v' = \frac{v - min_A}{max_A\_min_A}(new\_max_A - new\_min_A) + new\_min_A \tag{7}$$

v′ = new data, v = old data, $\min_A$ = the smallest value of an attribute, $\max_A$ = the biggest value of an attribute, new_min_A = the smallest value of a new attribute (=0), new_max_A = the biggest value of a new attribute (=1).

5.4 Determining Neural Network Topology

Changing network topology was done by varying the number of hidden layers by referring to the number of the components formed in PCA. The number of hidden neurons in each hidden layer is changed by trial and error.

[1] UCI Machine Learning Repository Homepage, https://archive.ics.uci.edu/ml/datasets/Forest+type+mapping, last accessed 2019/01/10.

5.5 Training and Testing

Training was carried out with 70% data from the dataset, while testing was carried out with 30% data from the dataset. The training and testing process would stop when the iteration had reached 500 times.

6 Result and Discussion

6.1 Principal Component Anaysis Result

By analyzing the main components of the input attributes on the dataset that was going to be used for neural network training, the following results were obtained (see Table 1):

Table 1. Principal component analysis processing results

Component	Total	% of variance	Cumulative %
1	10.146	37.578	37.578
2	5.014	18.571	56.149
3	3.772	13.971	70.120
4	1.980	7.333	77.453
5	1.501	5.560	83.013
6	1.328	4.919	87.932
7	.875	3.240	91.172
8	.621	2.301	93.473
9	.511	1.893	95.366
10	.368	1.365	96.731
11	.318	1.179	97.910
12	.176	.652	98.561
13	.112	.416	98.977
14	.069	.256	99.233
15	.066	.244	99.477
16	.036	.132	99.609
17	.030	.112	99.721
18	.028	.104	99.826
19	.022	.081	99.907
20	.011	.042	99.948
21	.007	.026	99.974
22	.003	.013	99.987
23	.002	.009	99.996
24	.001	.003	99.999
25	.000	.001	100.000
26	8.995E−5	.000	100.000
27	2.354E−5	8.718E−5	100.000

If a minimum total diversity of 70% is taken, the number of hidden layers is three, if 80% of the total diversity of hidden layers is five, and if the total diversity is 90%, the number of hidden layers is seven. The number of components that will be used as the basis for determining the number of hidden layers in a neural network is chosen from one to four.

6.2 Neural Network Training Result

Neural network training was done by using multilayer perceptron architecture with the number of hidden layers chosen from one to four, while the number of neurons for each hidden layer was determined by the researcher. With the number of hidden layers selected and the number of neurons in each hidden layer determined, the neural network architecture to be trained can be seen in the following Table 2.

Table 2. The architecture of neural network

Architecture	Number of hidden layer	Number of neuron
I	1	36
II	1	100
III	2	15, 8
IV	3	15, 10, 5
V	3	20, 10, 5
VI	4	20, 15, 10, 5

The researcher gave an example as an explanation of the architecture in the above table that if architecture IV is a multilayer perceptron with three hidden layers, the first hidden layer has 15 neurons, the second hidden layer has 10 neurons, and the third hidden layer has 5 neurons. The training and testing were carried out on each of the above architectures, and the following results were obtained (see Table 3):

Table 3. The accuracy of neural network

Architecture	Accuracy
I	90.45%
II	89.17%
III	87.90%
IV	92.36%
V	92.36%
VI	27.39%

For architecture IV, the accuracy obtained was 92.36% which means that from the test data used, about 7.64% of the output was classified as incorrect.

6.3 Discussion

The experiments carried out used several different network topologies, both in terms of the number of hidden layers and neurons in each hidden layer, each of which has a certain level of accuracy by using the Forest Type Mapping dataset. From the experiments performed, it was shown that by using one hidden layer by varying the number of neurons 36(I) and 100(II), the accuracy were 90.44% and 89.18%, respectively. In the architecture of one hidden layer, the addition of the number of neurons did not increase the accuracy. For the use of two hidden layers, architecture III had an accuracy rate of 87.89% that also did not improve the accuracy compared to the use of one hidden layer. By using three hidden layers, architecture IV and V had an accuracy rate of 92.36%, which means that the use of three hidden layers was more accurate than the use of one or two hidden layers. The use of four hidden layers, in architecture VI, decreased the accuracy.

The experiments showed that the use of three hidden layers provided the best level of accuracy. If it is associated with PCA result, where it is assumed that the number of hidden layers selected was in accordance with the number of the components selected in PCA, the cumulative variance of the selected component was around 70%. This result indicated that the cumulative variance classification needed to determine the number of components does not have to be close to 100%. The classification of a group of different outputs, but still within a certain range, will be grouped in the same classification. It would be different if the regression objective function of each different output was considered as a different value.

7 Conclusion and Future Work

The right number of hidden layers chosen in neural network would provide high learning accuracy. Increasing the number of hidden layers does not guarantee an increase in their accuracy. The determination of the number of hidden layers could be done by using the number of components produced on the PCA principal component analysis by considering the cumulative variance. For the classification function, the number of cumulative variances could be done with a cut off value of 70%. For future works, the researchers need to consider the way to determine the number of neurons in each hidden layer besides determining the number of hidden layers. In addition, determining the number of hidden layers by using the number of PCA components also needs to be applied for the regression objective function. In the research that has been done, each neural network topology chosen is only conducted one-time experiments, in future studies for each topology the experiment repetition must be performed for example at least ten times, then each topology is compared. Experiments also need to be done for different types of datasets.

References

1. Madhiarasan, M., Deepa, S.N.: A novel criterion to select hidden neuron numbers in improved back propagation networks for wind speed forecasting. Appl. Intell. **44**(4), 878–893 (2016). https://doi.org/10.1007/s10489-015-0737-z
2. Madhiarasan, M., Deepa, S.N.: Comparative analysis on hidden neurons estimation in multi layer perceptron neural networks for wind speed forecasting. Artif. Intell. Rev. **48**(4), 449–471 (2017). https://doi.org/10.1007/s10462-016-9506-6
3. Qolomany, B., Maabreh, M., Al-Fuqaha, A., Gupta, A., Benhaddou, D.: Parameters optimization of deep learning models using particle swarm optimization. In: 13th International Wireless Communications and Mobile Computing Conference (IWCMC), pp. 1285–1290 (2017). https://doi.org/10.1109/IWCMC.2017.7986470
4. Thomas, A.J., Petridis, M., Walters, S.D., Gheytassi, S.M., Morgan, R.E.: Eng. Appl. Neural Netw. **744**, 279–290 (2017). https://doi.org/10.1007/978-3-319-65172-9
5. Thomas, A.J., Walters, S.D., Gheytassi, S.M., Morgan, R.E., Petridis, M.: On the optimal node ratio between hidden layers: a probabilistic study. Int. J. Mach. Learn. Comput. **6**(5), 241–247 (2016). https://doi.org/10.18178/ijmlc.2016.6.5.605
6. Guliyev, N.J., Ismailov, V.E.: On the approximation by single hidden layer feedforward neural networks with fixed weights. Neural Netw. **98**, 296–304 (2018). https://doi.org/10.1016/j.neunet.2017.12.007
7. Nitta, T.: Resolution of singularities introduced by hierarchical structure in deep neural networks. IEEE Trans. Neural Netw. Learn. Syst. **28**(10), 2282–2293 (2017). https://doi.org/10.1109/TNNLS.2016.2580741
8. Koutsoukas, A., Monaghan, K.J., Li, X., Huan, J.: Deep-learning: investigating deep neural networks hyper-parameters and comparison of performance to shallow methods for modeling bioactivity data. J. Cheminform. **9**(1), 1–13 (2017). https://doi.org/10.1186/s13321-017-0226-y
9. Bunjongjit, S., Ngaopitakkul, A., Pothisarn, C., Jettanasen, C.: Improvement to reduce training time of back-propagation neural networks for discrimination between external short circuit and internal winding fault. In: International Conference on Information Science, Electronics and Electrical Engineering, Sapporo, pp. 614–618 (2014)
10. Chhachhiya, D., Sharma, A., Gupta, M.: Designing optimal architecture of neural network with particle swarm optimization techniques specifically for educational dataset. In: 7th International Conference on Cloud Computing, Data Science and Engineering - Confluence, Noida, pp. 52–57 (2017)
11. Lee, S., Ha, J., Zokhirova, M., Moon, H., Lee, J.: Background information of deep learning for structural engineering. Arch. Comput. Methods Eng. **25**(1), 121–129 (2018). https://doi.org/10.1007/s11831-017-9237-0
12. Tej, M.L., Holban, S.: Comparative study of clustering distance measures to determine neural network architectures. In: IEEE 12th International Symposium on Applied Computational Intelligence and Informatics (SACI), pp. 189–194 (2018)
13. Tej, M.L., Holban, S.: Determining optimal neural network architecture using regression methods. In: International Conference on Development and Application Systems (DAS), pp. 180–189 (2018)
14. Tej, M.L.: Neural network architecture through data mining techniques. In: Proceedings of the IIER International Conference, Bucharest, Romania (2017)

15. Lorenzo, P.R., Nalepa, J., Kawulok, M., Ramos, L.S., Pastor, J.R.: Particle swarm optimization for hyper-parameter selection in deep neural networks. In: Proceedings of the 2017 on Genetic and Evolutionary Computation Conference - GECCO 2017, vol. 8, pp. 481–488 (2017). https://doi.org/10.1145/3071178.3071208
16. Bergstra, J., Bengio, Y.: Random search for hyper-parameter optimization. J. Mach. Learn. Res. **13**, 281–305 (2012)
17. Larochelle, H., Erhan, D., Courville, A., Bergstra, J., Bengio, Y.: An empirical evaluation of deep architectures on problems with many factors of variation. In: ICML (2007)
18. Bengio, Y., Goodfellow, I., Courville, A.: Deep Learning. MIT Press, Cambridge (2016)
19. Mitchell, T.M.: Machine Learning. McGraw-Hill, Boston (1997)
20. Jollife, I.T.: Principal Component Analysis. Springer, New York (2002)
21. Di, W., Bhardwaj, A., Wei, J.: Deep Learning Essentials. Packt Publishing, Birmingham (2018)

Determining the Number of Hidden Layers in Neural Network by Using Principal Component Analysis

Muh. Ibnu Choldun R.(✉), Judhi Santoso, and Kridanto Surendro

School of Electrical Engineering and Informatics, Institut Teknologi Bandung, Jl. Ganecha 10, Bandung, Indonesia
ibnucholdun@poltekpos.ac.id, judhi@stei.itb.ac.id, endro@informatika.org

Abstract. One of the challenges faced in the success of Deep Neural Network (DNN) implementation is setting the values for various hyper-parameters, one of which is network topology that is closely related to the number of hidden layers and neurons. Determining the number of hidden layers and neurons is very important and influential in DNN learning performance. However, up to now, there has been no guidance on it. Determining these two numbers manually (usually through trial and error methods) to find fairly optimal arrangement is a time-consuming process. In this study, we propose the method used for determining the number of hidden layers was through the number of components formed on the principal component analysis (PCA). By using Forest Type Mapping Data Set, based on PCA analysis, it was found out that the number of hidden layers that provide the best accuracy was three. This is in accordance with the number of components formed in the principal component analysis which gave a cumulative variance of around 70%.

Keywords: Neural network · Hidden layer · PCA

1 Introduction

Neural network has been successfully applied in various fields such as computer science, finance, medicine, engineering, physics, and so forth. The main reason is that neural network could approach arbitrary function. Over the past 30 years, a number of researches showed that artificial neural network called as feedforward network with one hidden layer can approach all functions arbitrarily [1–6]. One of important aspects for designing neural network is architecture or its topology that is closely related to the capability of generalization [7, 8]. Since the era of deep learning, the use of neural network that has more than one hidden layer is a research topic that caught researchers' interest. One of the challenges in the success of deep neural network implementation is setting values for various hyper-parameters, one of which is network topology that is closely related to the number of hidden layers and neurons. Determining the number of hidden layers and neurons is very important and influential in the performance of deep neural network [7, 8]. Therefore research to determine both of them is very necessary.

© Springer Nature Switzerland AG 2020
Y. Bi et al. (Eds.): IntelliSys 2019, AISC 1038, pp. 490–500, 2020.
https://doi.org/10.1007/978-3-030-29513-4_36

According to some literatures aforementioned, determining the number of hidden layers or optimal number of neurons still has no clear guideline. Moreover, the roles and functions of both are explained minimally. Some literatures propose methods or ways to determine the number of hidden layers or neurons, but they are not generally applied due to the type of input and output data [9, 10]. Studies in this area still leave difficult research tasks [11]. Some of the methods mentioned are apparently not applicable for different types of data. The earlier researchers determined the number of hidden layers or neurons based on their experience. However, less experienced researchers even did it based on 'trial and error'. One of the methods that can be applied practically is through data mining techniques, clustering and regression [12–14]. In this study, the researcher did not explain the relationship between the number of clusters formed by feature extraction that is going to be used in machine learning process.

Although it is a difficult area of research, determining the number of hidden layers and neurons should be carried out. This is because they greatly determine the deep neural network learning performance. As a first step, a method or technique is needed to determine the number of hidden layers. Hidden layers in the neural network in stages represent increasingly complex features, while the main components of the PCA represent the information content that exists. By assuming that the complexity of features is in line with the size of the feature information, then in this study we propose a method specifically for determining the number of hidden layers in a neural network based on the number of main components formed on principal component analysis (PCA).

2 Related Work

Studies compared the use of one or two hidden layers focused on univariate and multivariate functions [4–6, 15]. Thomas [4, 5] got different result that the use of two hidden layers applied to predictive functions showed better performance. Guliyev and Ismailov [6] concluded that the use of one hidden layer was less capable of approaching the multivariate function, so the use of two hidden layers showed better performance.

To determine the number of hidden layers and neurons at once, the method is divided into two: automatically and manually [15]. Automatic determination uses models, but it also can be done without models. Manual determination uses certain formulas, some are done by trial and error. Automatic determination without a model consists of grid search (GS) and random search (RS). GS is generally used to optimize DNN parameters only if the number is very low. First, users choose a range of values to explore. Then, DNN is set for each specification along with other parameters. Usually, GS is carried out for logarithmic scale steps that the best combination is estimated progressively to search [16]. The obvious disadvantage of GS is its time complexity – parameter k takes different values (by assuming that n is the same for all parameters), the complexity grows exponentially at level $O(n^k)$. Random search (RS) is an alternative to GS. It integrates faster into parametricization. RS is not adaptive that it is not dynamically updated during the experiment (the solutions found did not do anything to

the search). Furthermore, there is also hybrid algorithm that combines RS with other techniques to improve its performance, for instance, manual updates provided by experts [16, 17]. The technique that has been done turns out to be not generally accepted.

Model-based algorithm builds replacement models for hyper-parameter space. Then, hyper-parameters are set by using this model. Most of these techniques use Bayesian regression model that turns this problem into a trade-off between exploration (traversing unknown areas of space that classification of the performance is unknown), and exploitation (analyzing hyper-parameters that are likely to perform well and closed to "good" hyper-parameters that have been investigated). One of model-based algorithms is the use Particle Swarm Optimization (PSO) [3, 15]. PSO is an optimization technique done by continuously calculating potential solutions by using a quality reference. This algorithm optimizes problems by moving particles or prospective solutions for problems through certain functions for the position and velocity of the particles. The particles' movement is influenced by the best solution for the particles and the best solution in general that is obtained from other particles. This set of particles is called as a swarm that in the end it will move towards the best solution.

Although some researchers have proposed methods to calculate the number of hidden layers and neurons at once, but the methods were rarely used by other researchers, because they are difficult to apply or do not match the type of dataset used, therefore some researchers decided to use trial and error methods [8]. Therefore, a method or technique is needed to calculate the number of hidden layers or hidden neurons that can be generally accepted.

3 Neural Network

Architecture of back propagation neural network consists of several layers: input layer, hidden layer and output layer. Each neuron in each layer is connected to the next layer. The neurons in the input layer are connected to neurons in the hidden layer through synapses called as weight. In addition, hidden layers of neurons connected through synapses with neurons in the output layer. One of the neural network architectures is called as Multilayer Perceptron (MLP) [18]. MLP consists of: input layers where the neurons are fully connected to each neuron in the first hidden layer, each neuron in the hidden layer is connected to each neuron in the next hidden layer, and each neuron in the last hidden layer is fully connected with each neuron in the output layer. MLP, or often called as feedforward deep network, is a classic example of deep learning model.

Figure 1 is an example of a neural network with an MLP architecture consisting of input layers, two hidden layers, and an output layer. The closer the hidden layer to the output layer the better it identifies the complex features. For instance, MLP architecture that uses three hidden layers for face identification: the first hidden layer is used to identify the geometric shape that can distinguish the facial parts, the second hidden layer is used to identify eyes, nose, mouth, etc., and the third hidden layer is used to identify the face.

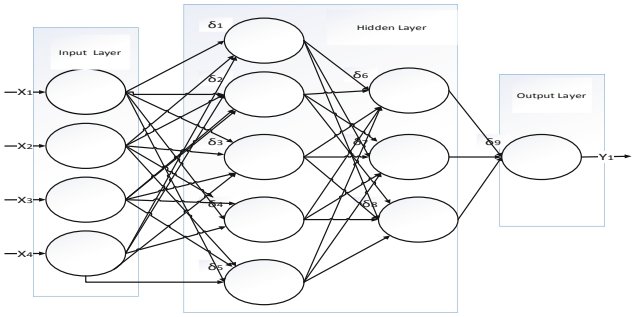


Fig. 1. Neural network

The training process for neural network is done by finding the value of each connection weight, which is done iteratively. Therefore, the output can correctly predict the target value of the training data examples. Backpropagation is an algorithm that is widely used for neural network learning, especially in MLP architecture. Each learning algorithm developed is based on inductive bias, a collection of assumptions that underlie the selection criteria of learning algorithm model. According to Mitchell [19], a learner's inductive bias is a set of additional assumptions that is sufficient to justify an inductive conclusion as a deductive conclusion. An example of an inductive bias that can be applied to MLP learning is to obtain the desired accuracy that the correct architecture must be chosen (the number of hidden layers and neurons for each hidden layer).

After obtaining the output from the perceptron based on the input given, at the error stage the perceptron calculation will be evaluated "whether the perceptron output matches the expected output or not". To calculate the difference between actual output and desired output can be used various functions of errors such as Squared Error (Eq. 1), Root Mean Squared Error (RMSE), or Cross Entropy (Eq. 2).

$$E = \frac{1}{2}\sum_{i=1}^{N}(t_i - y_i)^2 \tag{1}$$

for t = actual output, y = desired output, and N = number of data, then squared errors from t and y can be calculated using Eq. 1. The use of squared error often has problems if it is implemented in the case of classification with logistical output. Unlike the squared error, cross entropy does not experience problems in this case. In cross entropy, the greater the prediction error of the neural network, the greater the update value for the weight. The error cross entropy function can be formulated as follows:

$$E = -\frac{1}{N}\sum_{i=1}^{N} a_i \log(y_i) + (1 - a_i)\log(1 - y_i) \tag{2}$$

where y_i is the output of the neuron to i, a_i is the *actual output* to i, and N the total number of datasets.

4 Principal Component Analysis

Principal Component Analysis (PCA) is a multivariate analysis that transforms correlated origin variables into new variables that do not correlate with each other by reducing a number of the variables mentioned, so that the variables have smaller dimension but can explain most of the diversity of the origin variables [20]. The PCA steps are as follows.

1. There is a data matrix ($X = [x_1, x_2, \ldots, x_N]$), N is the total number of samples and xi represents the sample to i.
2. Calculate the average of all samples, as follows:

$$\mu = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{3}$$

3. Subtract all data samples with the average, as follows:

$$D = \{d_1, d_2, \ldots, d_N\} = \sum_{i=1}^{N} x_i - \mu \tag{4}$$

4. Calculate the covariance matrix, as follows:

$$\sum = \frac{1}{N-1} D \times D^T \tag{5}$$

5. Calculate the eigenvectors V and eigenvalues λ from the covariance matrix (Σ)
6. Sort the eigenvectors based on the eigenvalues that correspond
7. Select eigenvectors which have the largest eigenvalues $W = \{v_1, \ldots, v_k\}$. The selected Eigenvectors (W) represent the PCA projection space.
8. All samples are projected on the dimensional space lower than PCA (W), as follows:

$$Y = W^T D \tag{6}$$

The number of the main components formed is equal to the number of original variables. The selection of the main components used was based on the eigenvalue value that the main components will be used if the eigenvalue is greater than one. However, the reduction (simplification) of the dimension can also be done by finding the percentage criteria of the data diversity explained by the first few main components. For example, the number of the main components used was if the cumulative variance of the components was at least 70%.

5 Research Method

5.1 Research Framework

The more hidden layers in a neural network the more they can represent the increasingly complex features [19]. Deep architecture in neural network can detect features

arranged in each layer. The lower layers (that are closer to the input layer) could detect simpler features, while the layers closer to the ouput layer could detect more complex features [21]. Increasingly complex feature represents higher information content received. The high information in PCA is represented in the main component that has high variance. In other words, increasingly complex feature in hidden layers of the neural network is in accordance with the main components that have higher variance. As a result, the number of hidden layers in the required neural network will be in accordance with the number of the main components in the principal component analysis. Therefore, in this study, the number of hidden layers in the neural network could be determined based on the number of the main components formed through PCA.

5.2 Preparing the Dataset

The data set used in this study was Forest Type Mapping Data Set that was converted into numeric data. The data sets could be downloaded on UCI Machine Learning Repository website[1]. The number of input attribute was 27 and the output category was 4 classes: 's' ('Sugi' forest), 'h' ('Hinoki' forest), 'd' ('Mixed deciduous' forest), 'o' ('Other' non-forest land). The number of dataset used was 325 records, 70% of which was used as training data and the remaining is for testing. The output function used in the data set was classification.

5.3 Standardization and Normalization

Standardization was done by deleting data that had empty attributes and also eliminating extreme values. Normalization was done with the aim of getting data with attributes that are scaled accordingly. The normalization done was Min-Maks with a value between 0 and 1. Min-Maks normalization formula is as follows:

$$v' = \frac{v - min_A}{max_A\_min_A}(new\_max_A - new\_min_A) + new\_min_A \tag{7}$$

v′ = new data, v = old data, $\min_A$ = the smallest value of an attribute, $\max_A$ = the biggest value of an attribute, new_min_A = the smallest value of a new attribute (=0), new_max_A = the biggest value of a new attribute (=1).

5.4 Determining Neural Network Topology

Changing network topology was done by varying the number of hidden layers by referring to the number of the components formed in PCA. The number of hidden neurons in each hidden layer is changed by trial and error.

[1] UCI Machine Learning Repository Homepage, https://archive.ics.uci.edu/ml/datasets/Forest+type+mapping, last accessed 2019/01/10.

5.5 Training and Testing

Training was carried out with 70% data from the dataset, while testing was carried out with 30% data from the dataset. The training and testing process would stop when the iteration had reached 500 times.

6 Result and Discussion

6.1 Principal Component Anaysis Result

By analyzing the main components of the input attributes on the dataset that was going to be used for neural network training, the following results were obtained (see Table 1):

Table 1. Principal component analysis processing results

Component	Total	% of variance	Cumulative %
1	10.146	37.578	37.578
2	5.014	18.571	56.149
3	3.772	13.971	70.120
4	1.980	7.333	77.453
5	1.501	5.560	83.013
6	1.328	4.919	87.932
7	.875	3.240	91.172
8	.621	2.301	93.473
9	.511	1.893	95.366
10	.368	1.365	96.731
11	.318	1.179	97.910
12	.176	.652	98.561
13	.112	.416	98.977
14	.069	.256	99.233
15	.066	.244	99.477
16	.036	.132	99.609
17	.030	.112	99.721
18	.028	.104	99.826
19	.022	.081	99.907
20	.011	.042	99.948
21	.007	.026	99.974
22	.003	.013	99.987
23	.002	.009	99.996
24	.001	.003	99.999
25	.000	.001	100.000
26	8.995E−5	.000	100.000
27	2.354E−5	8.718E−5	100.000

If a minimum total diversity of 70% is taken, the number of hidden layers is three, if 80% of the total diversity of hidden layers is five, and if the total diversity is 90%, the number of hidden layers is seven. The number of components that will be used as the basis for determining the number of hidden layers in a neural network is chosen from one to four.

6.2 Neural Network Training Result

Neural network training was done by using multilayer perceptron architecture with the number of hidden layers chosen from one to four, while the number of neurons for each hidden layer was determined by the researcher. With the number of hidden layers selected and the number of neurons in each hidden layer determined, the neural network architecture to be trained can be seen in the following Table 2.

Table 2. The architecture of neural network

Architecture	Number of hidden layer	Number of neuron
I	1	36
II	1	100
III	2	15, 8
IV	3	15, 10, 5
V	3	20, 10, 5
VI	4	20, 15, 10, 5

The researcher gave an example as an explanation of the architecture in the above table that if architecture IV is a multilayer perceptron with three hidden layers, the first hidden layer has 15 neurons, the second hidden layer has 10 neurons, and the third hidden layer has 5 neurons. The training and testing were carried out on each of the above architectures, and the following results were obtained (see Table 3):

Table 3. The accuracy of neural network

Architecture	Accuracy
I	90.45%
II	89.17%
III	87.90%
IV	92.36%
V	92.36%
VI	27.39%

For architecture IV, the accuracy obtained was 92.36% which means that from the test data used, about 7.64% of the output was classified as incorrect.

6.3 Discussion

The experiments carried out used several different network topologies, both in terms of the number of hidden layers and neurons in each hidden layer, each of which has a certain level of accuracy by using the Forest Type Mapping dataset. From the experiments performed, it was shown that by using one hidden layer by varying the number of neurons 36(I) and 100(II), the accuracy were 90.44% and 89.18%, respectively. In the architecture of one hidden layer, the addition of the number of neurons did not increase the accuracy. For the use of two hidden layers, architecture III had an accuracy rate of 87.89% that also did not improve the accuracy compared to the use of one hidden layer. By using three hidden layers, architecture IV and V had an accuracy rate of 92.36%, which means that the use of three hidden layers was more accurate than the use of one or two hidden layers. The use of four hidden layers, in architecture VI, decreased the accuracy.

The experiments showed that the use of three hidden layers provided the best level of accuracy. If it is associated with PCA result, where it is assumed that the number of hidden layers selected was in accordance with the number of the components selected in PCA, the cumulative variance of the selected component was around 70%. This result indicated that the cumulative variance classification needed to determine the number of components does not have to be close to 100%. The classification of a group of different outputs, but still within a certain range, will be grouped in the same classification. It would be different if the regression objective function of each different output was considered as a different value.

7 Conclusion and Future Work

The right number of hidden layers chosen in neural network would provide high learning accuracy. Increasing the number of hidden layers does not guarantee an increase in their accuracy. The determination of the number of hidden layers could be done by using the number of components produced on the PCA principal component analysis by considering the cumulative variance. For the classification function, the number of cumulative variances could be done with a cut off value of 70%. For future works, the researchers need to consider the way to determine the number of neurons in each hidden layer besides determining the number of hidden layers. In addition, determining the number of hidden layers by using the number of PCA components also needs to be applied for the regression objective function. In the research that has been done, each neural network topology chosen is only conducted one-time experiments, in future studies for each topology the experiment repetition must be performed for example at least ten times, then each topology is compared. Experiments also need to be done for different types of datasets.

References

1. Madhiarasan, M., Deepa, S.N.: A novel criterion to select hidden neuron numbers in improved back propagation networks for wind speed forecasting. Appl. Intell. **44**(4), 878–893 (2016). https://doi.org/10.1007/s10489-015-0737-z
2. Madhiarasan, M., Deepa, S.N.: Comparative analysis on hidden neurons estimation in multi layer perceptron neural networks for wind speed forecasting. Artif. Intell. Rev. **48**(4), 449–471 (2017). https://doi.org/10.1007/s10462-016-9506-6
3. Qolomany, B., Maabreh, M., Al-Fuqaha, A., Gupta, A., Benhaddou, D.: Parameters optimization of deep learning models using particle swarm optimization. In: 13th International Wireless Communications and Mobile Computing Conference (IWCMC), pp. 1285–1290 (2017). https://doi.org/10.1109/IWCMC.2017.7986470
4. Thomas, A.J., Petridis, M., Walters, S.D., Gheytassi, S.M., Morgan, R.E.: Eng. Appl. Neural Netw. **744**, 279–290 (2017). https://doi.org/10.1007/978-3-319-65172-9
5. Thomas, A.J., Walters, S.D., Gheytassi, S.M., Morgan, R.E., Petridis, M.: On the optimal node ratio between hidden layers: a probabilistic study. Int. J. Mach. Learn. Comput. **6**(5), 241–247 (2016). https://doi.org/10.18178/ijmlc.2016.6.5.605
6. Guliyev, N.J., Ismailov, V.E.: On the approximation by single hidden layer feedforward neural networks with fixed weights. Neural Netw. **98**, 296–304 (2018). https://doi.org/10.1016/j.neunet.2017.12.007
7. Nitta, T.: Resolution of singularities introduced by hierarchical structure in deep neural networks. IEEE Trans. Neural Netw. Learn. Syst. **28**(10), 2282–2293 (2017). https://doi.org/10.1109/TNNLS.2016.2580741
8. Koutsoukas, A., Monaghan, K.J., Li, X., Huan, J.: Deep-learning: investigating deep neural networks hyper-parameters and comparison of performance to shallow methods for modeling bioactivity data. J. Cheminform. **9**(1), 1–13 (2017). https://doi.org/10.1186/s13321-017-0226-y
9. Bunjongjit, S., Ngaopitakkul, A., Pothisarn, C., Jettanasen, C.: Improvement to reduce training time of back-propagation neural networks for discrimination between external short circuit and internal winding fault. In: International Conference on Information Science, Electronics and Electrical Engineering, Sapporo, pp. 614–618 (2014)
10. Chhachhiya, D., Sharma, A., Gupta, M.: Designing optimal architecture of neural network with particle swarm optimization techniques specifically for educational dataset. In: 7th International Conference on Cloud Computing, Data Science and Engineering - Confluence, Noida, pp. 52–57 (2017)
11. Lee, S., Ha, J., Zokhirova, M., Moon, H., Lee, J.: Background information of deep learning for structural engineering. Arch. Comput. Methods Eng. **25**(1), 121–129 (2018). https://doi.org/10.1007/s11831-017-9237-0
12. Tej, M.L., Holban, S.: Comparative study of clustering distance measures to determine neural network architectures. In: IEEE 12th International Symposium on Applied Computational Intelligence and Informatics (SACI), pp. 189–194 (2018)
13. Tej, M.L., Holban, S.: Determining optimal neural network architecture using regression methods. In: International Conference on Development and Application Systems (DAS), pp. 180–189 (2018)
14. Tej, M.L.: Neural network architecture through data mining techniques. In: Proceedings of the IIER International Conference, Bucharest, Romania (2017)

15. Lorenzo, P.R., Nalepa, J., Kawulok, M., Ramos, L.S., Pastor, J.R.: Particle swarm optimization for hyper-parameter selection in deep neural networks. In: Proceedings of the 2017 on Genetic and Evolutionary Computation Conference - GECCO 2017, vol. 8, pp. 481–488 (2017). https://doi.org/10.1145/3071178.3071208
16. Bergstra, J., Bengio, Y.: Random search for hyper-parameter optimization. J. Mach. Learn. Res. **13**, 281–305 (2012)
17. Larochelle, H., Erhan, D., Courville, A., Bergstra, J., Bengio, Y.: An empirical evaluation of deep architectures on problems with many factors of variation. In: ICML (2007)
18. Bengio, Y., Goodfellow, I., Courville, A.: Deep Learning. MIT Press, Cambridge (2016)
19. Mitchell, T.M.: Machine Learning. McGraw-Hill, Boston (1997)
20. Jollife, I.T.: Principal Component Analysis. Springer, New York (2002)
21. Di, W., Bhardwaj, A., Wei, J.: Deep Learning Essentials. Packt Publishing, Birmingham (2018)

This author profile is generated by Scopus. Learn more

Rachmatullah, Muhammad Ibnu Choldun

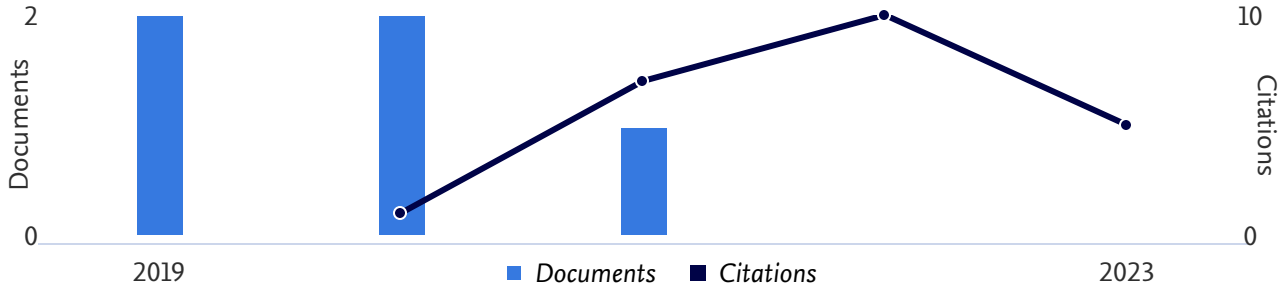
Institut Teknologi Bandung, Bandung, Indonesia 57224844807

https://orcid.org/0000-0003-4071-887X

23	5	3
Citations by ***20*** documents	Documents	*h*-index View *h*-graph

Set alert Edit profile More

Document & citation trends



Scopus Preview

Scopus Preview users can only view a limited set of features. Check your institution's access to view all documents and features.

Check access

Documents 1 Preprint 4 Co-Authors Topics Beta 0 Awarded Grants

Documents (5) Cited by (20)

Note:

Scopus Preview users can only view an author's last 10 documents, while most other features are disabled. Do you have access through your institution? Check your institution's access to view all documents and features.

5 documents

Export all Add all to list Sort by Date (...

Article • *Open access*

Determining the number of hidden layer and hidden neuron of neural network for wind speed prediction

2 Citations

Rachmatullah, M.I.C., Santoso, J., Surendro, K.

PeerJ Computer Science, 2021, 7, pp. 1–19

Show abstract Related documents

Article • *Open access*

A novel approach in determining neural networks architecture to classify data with large number of attributes

3 Citations

Rachmatullah, M.I.C., Santoso, J., Surendro, K.

IEEE Access, 2020, 8, pp. 204728–204743

Show abstract Related documents

Conference Paper

Determining the number of hidden layers in neural network by using principal component analysis

13 Citations

Ibnu Choldun R., M., Santoso, J., Surendro, K.

Advances in Intelligent Systems and Computing, 2020, 1038, pp. 490–500

Show abstract Related documents

Conference Paper

Determining the Neural Network Topology from the Viewpoint of Kuhn's Philosophy and Popper's Philosophy

0 Citations

Rachmatullah, M.I.C., Surendro, K., Santoso, J., Mahayana, D.

Proceeding - 2019 International Conference of Artificial Intelligence and Information Technology, ICAIIT 2019, 2019, pp. 115–118, 8834664

Show abstract Related documents

Conference Paper

Determining the neural network topology: A review

5 Citations

Muh Ibnu Choldun, R., Santoso, J., Surendro, K.

ACM International Conference Proceeding Series, 2019, Part F147956, pp. 357–362

Show abstract Related documents

Back to top

View list in search results format

View references

Set document alert

About Scopus

What is Scopus

Content coverage

Scopus blog

Scopus API

Privacy matters

Language

日本語版を表示する

查看简体中文版本

查看繁體中文版本

Просмотр версии на русском языке

Customer Service

Help

Tutorials

Contact us

ELSEVIER

Terms and conditions Privacy policy

Copyright © Elsevier B.V . All rights reserved. Scopus® is a registered trademark of Elsevier B.V.
We use cookies to help provide and enhance our service and tailor content. By continuing, you agree to the use of cookies .

RELX