1862
RIGA TECHNICAL
UNIVERSITY

Andrejs Bondarenko

DEVELOPMENT OF KNOWLEDGE EXTRACTION METHODOLOGY FROM TRAINED ARTIFICIAL NEURAL NETWORKS

Summary of the Doctoral Thesis

RTU Press
Riga 2020

RIGA TECHNICAL UNIVERSITY
Faculty of Computer Science and Information Technology
Institute of Information Technology

Andrejs Bondarenko
Doctoral Student of the Study Programme "Information Technology"

DEVELOPMENT OF KNOWLEDGE EXTRACTION METHODOLOGY FROM TRAINED ARTIFICIAL NEURAL NETWORKS

Summary of the Doctoral Thesis

Scientific supervisors:

Professor Dr. habil. sc. comp.
ARKĀDIJS BORISOVS

Professor Dr. sc. ing.
LUDMILA ALEKSEJEVA

RTU Press
Riga 2020

Bondarenko, A. Development of Knowledge Extraction Methodology From Trained Artificial Neural Networks. Summary of the Doctoral Thesis. Riga: RTU Press, 2020. 44 p.

Published in accordance with the decision of the Institute of Information Technology Council of 25 November 2019, Minutes No. 12100-2/8.

ISBN 978-9934-22-011-1 (print)
ISBN 978-9934-22-140-8 (pdf)

DOCTORAL THESIS PROPOSED TO RIGA TECHNICAL UNIVERSITY FOR THE PROMOTION TO THE SCIENTIFIC DEGREE OF DOCTOR OF SCIENCES

To be granted the scientific degree of Doctor of Sciences (Ph. D.), the present Doctoral Thesis has been submitted for the defence at the open meeting of RTU Promotion Council on 18 May 2020 at the Faculty of Computer Science and Information Technology of Riga Technical University, 1 Setas Street, Room 202.

OFFICIAL REVIEWERS

Professor Dr. habil. sc. ing. Jānis Grundspeņķis
Riga Technical University

Professor Dr. sc. ing. Pēteris Grabusts
Rezekne Academy of Technologies, Latvia

Professor Dr. habil. sc. ing. Yevgeniy Bodyanskiy
Kharkiv National University of Radio Electronics, Ukraine

DECLARATION OF ACADEMIC INTEGRITY

I hereby declare that the Doctoral Thesis submitted for the review to Riga Technical University for the promotion to the scientific degree of Doctor of Sciences (Ph. D.) is my own. I confirm that this Doctoral Thesis had not been submitted to any other university for the promotion to a scientific degree.

Andrejs Bondarenko ……………………………. (signature)
Date: ………………………

The Doctoral Thesis has been written in English. It consists of an introduction, 5 chapters, conclusions, 37 figures, 22 tables, 5 appendices; the total number of pages is 158, including appendices. The Bibliography contains 150 titles.

TABLE OF CONTENTS

GENERAL DESCRIPTION OF THE WORK

Introduction

Artificial neural networks (ANN) are widely used in machine learning. They are powerful non-linear models that can be trained in a supervised, semi-supervised, and unsupervised manner. There is no single best machine learning classifier that can be used in all scenarios, but ANNs are frequently outperforming other classifiers. On the downside, it is hard to explain how classification decision is made within ANN. Artificial neural networks are essentially black-boxes. Lack of understanding of how such classifiers work severely limits their applicability. The Thesis is devoted to the development of approaches allowing to extract knowledge in the form of rules from trained ANN classifier.

Topicality

Comprehensibility of the classification model is a crucial requirement in mission-critical domain areas like nuclear power, medicine, finance, and others. Additionally there could be law requirements, like the European Union GDPR 2018 law [118] stating that all life-changing algorithmic decisions should be explainable. Explainability allows ensuring there are no classification biases and discrimination and can generate new knowledge. There exist publications in the knowledge extraction domain, but no ready to use algorithms are available. In addition, as it was discovered reproducibility is a huge problem, thus the development of tooling for explaining ANN classifiers can significantly improve their usability.

Research Aim and Tasks

The research **aim** is to develop algorithms for pruning and knowledge extraction (KE) from trained ANN and unify them into knowledge extraction methodology. Such methodology should allow representing trained feedforward neural networks as an If–Then rules set, as a binary classification tree or set of equation rules. Research **tasks** to be solved to accomplish the stated research aim are the following.

1. To review and analyse existing knowledge representation and extraction approaches described in scientific literature addressing the same problem.
2. To study artificial neural networks pruning methods, their pros and cons, develop an improved method and evaluate it.
3. To develop, implement and evaluate approaches, which allow performing knowledge extraction from trained multilayer perceptron.
4. To develop and assess optimization-based methods for If–Then and elliptical rules extraction from trained piece-wise linear classifier and RBFNN.
5. To develop a generalized methodology for knowledge extraction from ANN.

Research Object and Subject

The research **object** is an explanation of trained artificial neural network classification decision, research **subject** – machine learning and specifically knowledge extraction approaches.

Research Hipotheses

During research and development of ANN pruning techniques and rules extraction methods, the following **hypotheses** were defined.

1. Improved sensitivity-based pruning algorithm successfully escapes local minimums and controls classification error rise.
2. Discreet input space subdivision acquired from MLP neurons outputs can be used to build a classification decision tree with controllable classification precision.
3. If–Then rules acquired via posing and solving the convex optimization problem allow approximate input space regions bounded by hyperplanes.
4. Elliptical rules extracted from feedforward radial basis function neural network by solving a non-convex optimization problem allow to approximate original RBFNN.

Research Methods

The study is based on mathematical and statistical analysis, machine learning, optimization theory, and experimental research methodologies. Literature review and analysis are used as well to gather information about the existing approaches in the subject domain area.

Scientific Novelty of the Thesis

The **scientific novelty** of the study is based on reviewing the existing and developing methods for knowledge extraction. Which, in turn, holds four specifically developed methods, which can be applied whenever model understanding and knowledge in explicit form is required. The scientific novelty and achievements are listed below.

1. A sensitivity-based artificial neural network pruning algorithm developed with several modifications allowing it to escape local minimums. A comparison of weights versus nodes pruning performed with recommendations on approach selection depending on requirements.
2. An implemented method for binary classification decision tree extraction from trained feedforward ANN multi-class classifier. An experimental testing of the proposed solution was performed.
3. An approach developed for extraction of If–Then rules from piece-wise linear approximation of a non-linear classifier. This approach allows rules extraction from a set of hyperplanes defined in input space. Although the approach has shown itself as

prone to curse of dimensionality, it can be utilized for datasets with a small amount of input data dimensions.

4. An approach developed for extraction of elliptical rules from two-dimensional or three-dimensional RBF neural network. Although this approach has shown itself prone to the curse of dimensionality developed optimization problem posed as the right approach for elliptical rules extraction in case of two or three dimensions. Larger dimensions counts can be supported via algorithmic enhancement.
5. Based on the conducted research and experiments developed the methodology for utilization of rules extraction approaches, listed above, with recommendations on cases when one approach should be selected over the other.

Practical Significance of Thesis

The practical significance lies in programmatic realizations, experimental validation and assessment of the discussed methods. Full list of practical achievements is as follows.

1. Performed review and comparison of knowledge representation schemas, and recommendations are given for scheme selection.
2. In the scope of developed methodology, recommendations allowing extraction of accurate or comprehensible rules are given.
3. Recommendations for using nodes or weights pruning are given, allowing to rise generalization of ANN.
4. Applicability of reviewed classifiers (MLP, RBFNN, Piece-Wise linear classifier) has been improved, as now it becomes possible to validate them, understand model classification decision, and discover new knowledge.
5. Programmatic realizations have been created in Matlab, extensions of Lua-based Torch7 deep learning (DL) framework and Python-based PyTorch DL framework. PyTorch version is applicable to medium-sized datasets.

Approbation

Research results were presented at thirteen international scientific conferences.

1. RTU 60th International Scientific Conference. Latvia, Riga, 10–11 October 2019.
2. RTU 59th International Scientific Conference. Latvia, Riga, 10–12 October 2018.
3. RTU 57th International Scientific Conference. Latvia, Riga, 17–21 October 2016.
4. 10th International Scientific and Practical Conference “Environment. Technology. Resources”. Latvia, Rezekne, 18–20 June 2015.
5. RTU 55th International Scientific Conference. Latvia, Riga, 14–16 October 2014.
6. 6th International Conference “Applied Information and Communication Technology”. Latvia, Jelgava, 25–26 April 2013.
7. RTU 53rd International Scientific Conference. Latvia, Riga, 11–12 October 2012.
8. International Conference “Information Intelligent Systems”. Ukraine, Kharkiv, 17–19 April 2012.

9. 10th WSEAS International Conference on Artificial Intelligence, Knowledge Engineering and Data Bases (AIKED'11). United Kingdom, Cambridge, 20–22 February 2011.
10. 17th International Conference on Soft Computing MENDEL. Czech Republic, Brno, 15–17 June 2011.
11. 15th International Conference on Knowledge-Based and Intelligent Information and Engineering Systems (KES 2011). Germany, Kaiserslautern, 12–14 September 2011.
12. 16th International Conference on Soft Computing MENDEL'10. Czech Republic, Brno,
    23–25 June 2010.
13. RTU 51st International Scientific Conference, Latvia, Riga, 11–15 October 2010.

Research results that served as the basis for the Thesis were published in the following scientific papers.

1. Bondarenko, A. Controlling Complexity and Accuracy of Classification Decision Tree Extracted from Trained Artificial Neural Network. In: 60*th International Scientific Conference on Information Technology and Management Science of Riga Technical University (ITMS), 2019*. Available from: doi:10.1109/ITMS47855.2019.8940739. **Indexed in Scopus.**
2. Bondarenko, A., Aleksejeva, L. Methodology for Knowledge Extraction from Trained Artificial Neural Networks. *Information Technology and Management Science*. 2018, vol. 21, pp. 6–14. Available from: doi:10.7250/itms-2018-001.
3. Bondarenko, A., Aleksejeva, L. Workflow for Knowledge Extraction from Neural Network Classifiers. In: *59th International Scientific Conference on Information Technology and Management Science of Riga Technical University (ITMS), 2018*. Available from: doi:10.1109/ITMS.2018.8552964. **Indexed in Scopus.**
4. Bondarenko, A., Aleksejeva, L., Jumutcs, V., Borisovs, A. Classification Tree Extraction from Trained Artificial Neural Networks. *Procedia Computer Science,* 2017, vol. 104, pp. 556–563. Available from: doi:10.1016/j.procs.2017.01.172. **Indexed in Scopus, Web of Science. Cited: 9.**
5. Bondarenko, A., Borisovs, A., Aleksejeva, L. Neurons vs Weights Pruning in Artificial Neural Networks. In: *Environment. Technology. Resources: Proceedings of the 10th International Scientific and Practical Conference, Latvia, Rezekne, 18–20 June 2015*. Vol. 3. Rezekne: Rezekne Higher Education Institution, 2015, pp. 22–28. Available from: doi:10.17770/etr2015vol3.166, **Indexed in Scopus. Cited: 2.**
6. Bondarenko, A., Borisovs, A. Artificial Neural Network Generalization and Simplification via Pruning. *Information Technology and Management Science.* 2014, vol. 17, pp. 132–137. Available from: doi:10.1515/itms-2014-0020.
7. Bondarenko, A., Borisovs, A. Elliptical Rule Extraction from a Trained Radial Basis Function Neural Network. In: *The 6th International Conference "Applied Information and Communication Technology" (CD-ROM)*, Latvia, Jelgava, LUA Faculty of Information Technology, 25–26 April 2013. **Indexed in Web of Science. Cited: 1.**

8. Bondarenko A., Borisov A. Research on the Classification Ability of Deep Belief Networks on Small and Medium Datasets. *Scientific Journal of Riga Technical University, Information Technology and Management Science*, 2013, vol. 16, pp. 60–65. Available from: doi:10.2478/itms-2013-0009. **Indexed in** EBSCO, De Gruyter, Google Scholar, ResearchGate. **Cited: 2.**
9. Bondarenko, A., Borisovs, A. Knowledge Extraction from Piecewise-Linear Approximation of Multi-Surface Classifier. In: *International Conference "Information Intelligent Systems", Kharkov, Ukraine, 17–19 April 2012*. Vol. 6, pp. 5–6.
10. Bondarenko, A., Borisov, A. The Extraction of Elliptical Rules from the Trained Radial Basis Function Neural Network. *Information Technology and Management Science*. 2012, vol. 15, pp. 161–165. Available from: doi:10.2478/v10313-012-0027-2.
11. Bondarenko, A., Jumutc, V. Extraction of Interpretable Rules from Piecewise-Linear Approximation of a Nonlinear Classifier using Clustering-Based Decomposition. *Proceedings of the 10th WSEAS international conference on Artificial intelligence, knowledge engineering and data bases (AIKED'11), United Kingdom, Cambridge, 22–22 February 2011*. Cambridge: 2011, pp.145–149. **Indexed in Scopus.**
12. Bondarenko, A., Zmanovska, T., Borisovs, A. Piece-Wise Classifier Application to RBF Neural Network Rules Extraction. In: *17th International Conference on Soft Computing (MENDEL'11), Czech Republic, Brno, 15–17 June 2011*. Brno: Brno University of Technology, 2011, pp. 170–176. **Indexed in Scopus, Web of Science**.
13. Jumutcs, V., Bondarenko, A. Polytope Classifier: A Symbolic Knowledge Extraction from Piecewise-Linear Support Vector Machine. In: *Knowledge-Based and Intelligent Information and Engineering Systems: 15th International Conference (KES 2011): Proceedings, Part 1, Germany, Kaiserslautern, 12–14 September 2011*. Berlin: Springer Berlin Heidelberg, 2011, pp. 62–71. Available from: doi:10.1007/978-3-642-23851-2_7. **Indexed in Scopus, Web of Science,** ResearchGate, SpringerLink.
14. Bondarenko, A., Borisov, A. Decompositional Rules Extraction Methods from Neural Networks. In: *Proceedings of the 16th International Conference on Soft Computing* MENDEL'10*, Czech Republic, Brno, 23–25 June 2010, Brno: University of Technology*, 2010, pp. 256–262. ISBN 9788021441200. **Indexed in Scopus, Web of Science**.
15. Bondarenko, A., Borisov, A. Research of Artificial Neural Networks Abilities in Printed Words Recognition. *Scientific Journal of Riga Technical University, Information Technology and Management Science*. 2010, vol. 44, issue 5, pp. 124–129.

The results of the Doctoral Thesis research have been used in the following projects.

1. Latvian Council of Science project No. 09.1564 "Simulation and computational intelligence methods for logistics and e-business optimization" (2010–2012). Project leader Professor Dr. habil. sc. ing. Y. Merkurjevs.
2. Latvian Council of Science funded project LZP-2018/2-0051 "Fast and non-contact optical estimation of micro-organisms activity" (2018−2020). Project leader Dr. phys. A. Lihačov.

3. Latvian Council of Science funded project LZP-2018/2-0052 “Skin cancer early diagnostics accuracy improvement by using neural networks” (2018–2020), Project leader Dr. phys. I. Lihačova.

Structure and Content of the Thesis

The Doctoral Thesis contains an introduction, five main chapters, results analysis, and conclusions.

The introduction validates the topicality of the conducted investigations, formulates the object, the aim, and research tasks. It describes scientific novelty, as well as briefly characterizes basic directions of the research performed.

Chapter 1 describes the problem area, which is artificial neural networks. A short introduction is given along with brief descriptions of the main well-known ANN types.

Chapter 2 describes in detail the initial step required for knowledge extraction from a trained artificial neural network – network pruning. This chapter covers several existing algorithms along with their evaluations and comparison to the developed pruning algorithm.

Chapter 3 presents the developed approach for the extraction of binary classification decision tree from a trained multilayer perceptron. This chapter describes the implemented algorithm, provides its pseudocode, and holds algorithm evaluation.

Chapter 4 presents the developed optimization-based methods for oblique (If–Then) and equation rules extraction from a piece-wise linear classifier and RBFNN.

Chapter 5 describes the developed methodology for choosing one of the described methods over others.

Results and Conclusions chapter recaps the aim, tasks, and hypotheses, makes conclusions, covers scientific and practical novelty of the Thesis, and discusses future research directions.

SUMMARY OF THESIS CHAPTERS

1. NEURAL NETWORKS AND KNOWLEDGE EXTRACTION

Based on the posed aim and tasks, the first chapter performs an overview of the machine-learning field and artificial neural networks (ANN) as a research object. This chapter covers ANN types, their training and usage. Knowledge extraction (KE) algorithms and knowledge representation forms are reviewed.

1.1. Neural Networks and Knowledge Extraction Overview

Look into [90] for the in-depth introduction to biological and artificial neural networks. Authors of [40] posed a plausible explanation describing how neural networks can operate and approximate simple linear functions. Later, researchers inspired by biological neural networks have proposed different artificial neural network architectures.

Data itself represent little value. Information on the other side is data bound into a specific context, which gives some meaning to the data and allows us to see relations. Knowledge is information, which is organized in a way that allows seeing specific patterns. Wisdom is even more abstract and allows an understanding of general principles. In the scope of general AI systems, knowledge had to be represented in such a way that specific reasoning could be applied over it. Knowledge is the level covered by current research, it uses the definition for "rule extraction" term proposed in [73]. Given definition is broader than others like one given in [96] and underlines the fact that extracted rules can take different forms, not only lexical:

> "Given an opaque predictive model and the data on which it was trained, produce a description of the predictive model's hypothesis that is understandable yet closely approximates the predictive model's behaviour."

Depending on the context, rule extraction algorithms can be tuned to produce either more comprehensible, and hence compact and understandable, or more accurate rules.

According to the knowledge extraction taxonomy proposed in [76], there exist three types of knowledge extraction algorithms. The fourth knowledge extraction algorithm family (compositional) was proposed in [139]. These types are decompositional, pedagogical, eclectic, and compositional.

Depending on the use-case, the decision regarding the trade-off between readability and high classification rate should be made. This decision will allow selecting the most appropriate rules type. Types of rules (see detailed description in [22], [73]) are: propositional If−Then / If−Then−Else rules; M of N rules; Oblique rules / Equational rules; and Fuzzy rules.

If–Then rules and Decision trees are most welcome as they are easily embeddable, have good comprehensibility and acceptable expressiveness and compactness. Equational and oblique rules are most expressive while least interpretable by a human expert. The last two groups – Fuzzy rules and M of N rules − are of less interest as they either bound to fuzzy neural networks, which are less common and are not easily embeddable, or have low expressive power. The current Thesis is focused on If–Then rules and binary classification decision tree extraction from MLP, as well as on extraction of elliptical rules from RBFNN.

As it was noted in [49], to get compact FFNN, there exist several approaches. The current Thesis is concentrating on the pruning approach. In case one does not have a trained neural network, the same pruning approach can be used for training an overly complex network with subsequent pruning to remove unnecessary neurons. Lowering the number of neurons can result in a smaller number of rules. The pruning algorithm developed in the scope of the Thesis is covered in Chapter 2.

1.2. Types of Knowledge Extraction Methods

There are four main types of KE algorithms, their strengths and weaknesses are summarized in Table 1.1. The current Thesis concentrates on decompositional and compositional KE algorithms. Decompositional algorithms, as it was shown in [39], are performing better than pedagogical approaches. Classification accuracy, portability and ability to influence the complexity and precision of extracted rules were selected as the most important properties.

The pedagogical approach is applied to RBFNN to extract Elliptical rules due to their high expressive power. For a decompositional approach If–Then rules and classification decision tree were chosen as knowledge representation forms to be explored. They are most commonly widespread, can be easily embedded into any existing information system and have an "embedded" inference engine, while being easily understandable.

Table 1.1

Comparison of Types of Knowledge Extraction Algorithms

Property	Algorithm class			
	Decompositional	Eclectic	Pedagogical	Compositional
Classification accuracy	++	+	+	n/a
Portability (not specific to classifier)	−	+−	+	−
Tunability	++	+	+	−
Algorithm consistency (several runs – same result)	−	+	+−	n/a
Speed	−	+	+	n/a
Knowledge representation variety	++	+	+	−
Scalability (Big data)	+	+	+	n/a
Algorithm complexity (computational)	−	+	+	+

The chapter provides a general overview of the ML field and ANN generalization theory. Knowledge representation schemas and typical knowledge extraction workflow are covered. Task number one of the stated research tasks is accomplished.

2. NEURAL NETWORK PRUNING

It is known that neural networks with a smaller number of neurons are easier to extract knowledge from, and extracted knowledge is of smaller complexity. The pruning step is always welcome as it can positively influence ANN generalization abilities. In addition, pruning controls the comprehensibility and accuracy of extracted rules by controlling the number of neurons to be processed. Papers [3], [6], [25] give an overview of pruning algorithms. To select the pruning approach, based on a literature review a summary Table 2.1 was created using the scale from zero to five (higher is better).

Table 2.1

Comparison of Pruning Approaches

Criteria / Pruning type	**Sensitivity based**	**Sensitivity analysis II OBD/OBS**	**Magnitude based**	**Weight decay**	**Mutual information-based**	**Significance based**	**Interactive pruning**
Simplicity	4	1	5	2	3	1	2
Execution time	0	1	3	2	3	1	1
Memory footprint	3	0	3	3	1	2	2
No special training procedure	2	1	2	0	2	2	2
Classification precision / generalization	3	2	0	n/a	n/a	3	n/a
Pruned neurons / weights count	3	3	0	n/a	n/a	4	n/a

Sensitivity-based pruning approach was chosen due to its simplicity and good reported performance. Its idea is to remove single neuron and assess the ANN performance change, thus the least sensitive (important) neuron can be found and removed.

2.1. Sensitivity-Based Pruning Algorithm

Experiments with basic sensitivity-based pruning approach have shown that it is prone to local minimums, thus improved algorithm version has been developed. To escape local minimums, the algorithm [26] has been equipped with three improvements. The main steps of the developed algorithm are:

1) save the ANN state;
2) determine the least sensitive neuron (or weight), remove it;
3) retrain ANN;
4) assess classification performance degradation; if it is acceptable, continue pruning, otherwise rollback last neuron (or weight) removal; if several consecutive rollbacks have occurred (threshold is reached), then stop pruning and return the last best known saved ANN.

Steps 1, 3 and 4 are the improvements proposed within the scope of the Thesis. They allowed to prune more neurons (or weights) and control classification degradation. To validate the proposed improvements, a pruning process visualization was created, see Figure 2.1

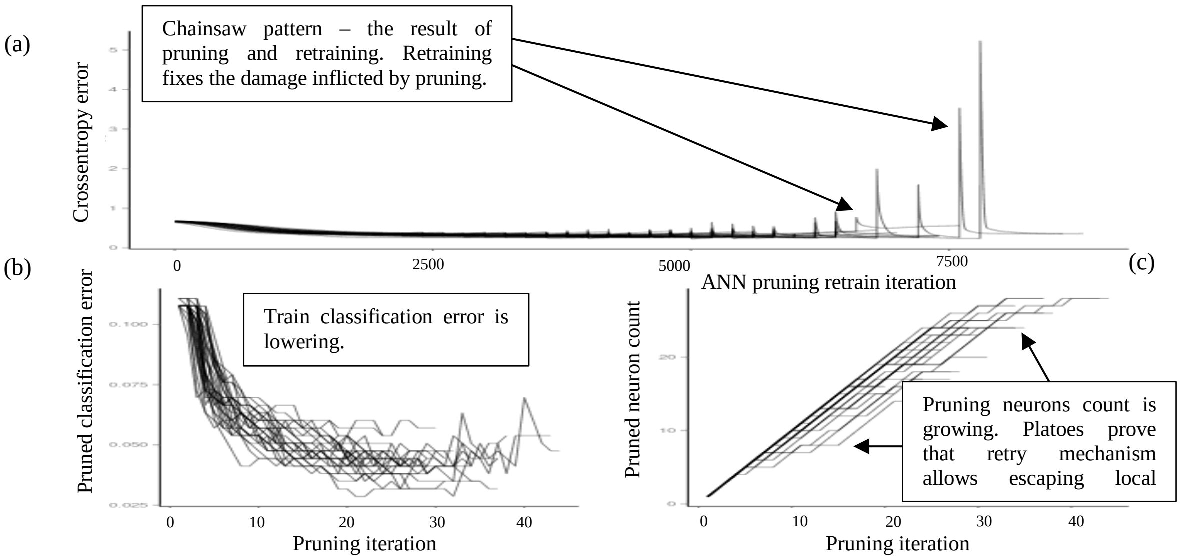


Fig. 2.1. Neuron pruning experiments for the Ionosphere dataset.

2.2. Validation of the Developed Pruning Algorithm

To validate the developed algorithm, an experiments plan has been developed. The goal was to validate the developed algorithm and compare neurons vs. weights pruning. In the first experiments series (Table 2.2), nodes pruning algorithm was applied to neurons in hidden layers only. It is seen, that in all but three cases the testing error of pruned ANN was smaller than that of unpruned ANN. The developed pruning algorithm has shown strong performance both in terms of ANN simplification and its generalization improvement.

Table 2.2

Results of Pruning Experiments

Dataset	**MLP train avg.**	**MLP test avg.**	**Pruned train**	**Pruned test avg.**	**Architecture before/after pruning (hidden nodes in 2 hidden layers)**
Ionosphere	10.83 %	10.83 %	5.39 %	**10.44** %	15–15 / 5.4–3.8
Monks-1	20.16 %	29.68 %	18.47 %	**24.35** %	15–15 / 5–3
Monks-2	36.82 %	36.55 %	31.83 %	**32.58** %	15–10 / 5–3.1
Monks-3	6.64 %	**2.80** %	5.98 %	2.85 %	15–15 / 1.7–1.1
WPBC	0.00 %	0.00 %	0.00 %	**0.00** %	10–10 / 1–1
WDBC	3.89 %	4.04 %	3.03 %	**3.69** %	30–30 / 23.2–17.1
Pima	23.02 %	**23.56** %	25.94 %	26.81 %	10–10 / 2.8–3.1
Haberman	26.13 %	**26.57** %	28.50 %	28.20 %	15–15 / 2.3–3.7
Parkinsons	24.62 %	24.61 %	16.29 %	**15.82** %	30–30 / 26.8–28

Experiments were conducted to validate the developed algorithm in application to nodes and weights pruning [25] applied to the input and hidden neurons, see Table 2.3 for results.

In Table 2.2 and Table 2.3 one can see that pruned ANN in two cases has a minor rise in classification error and in the majority of cases its error rate is lowered. In four cases (Table 2.3), weights pruning was better than nodes pruning. In the remaining five experiment sets, nodes pruning proved to be a better option. Error after weights pruning on the Monks-1 dataset is 1.81 % in contrast to 13.22 % after nodes pruning, this is the only case with such drastic difference.

Table 2.3

Classification Error Rates and Mean Pruned Weights / Nodes Counts

Dataset	MLP train avg.	MLP test avg.	Pruned weights train avg.	Pruned weights test avg.	Pruned nodes train avg.	Pruned nodes test avg.	Pruned weights / pruned nodes counts
Haberman	25.99 %	26.78 %	24.39 %	**24.91** %	24.98 %	26.17 %	54.9/23.8
Ionosphere	10.83 %	10.83 %	4.21 %	10.25 %	4.55 %	**9.22 %**	34.1/34.3
Monks-1	21.51 %	32.74 %	0.83 %	**1.81 %**	6.83 %	13.22 %	45.9/22.4
Monks-2	38.46 %	36.04 %	12.47 %	12.21 %	11.26 %	**10.25 %**	16.8/20.1
Monks-3	6.56 %	**2.88 %**	5.16 %	3.45 %	3.33 %	5.76 %	32.4/29.3
Parkinsons	24.58 %	24.61 %	14.83 %	16.38 %	14.30 %	**15.57 %**	10.5/8.3
Pima	23.93 %	24.56 %	21.64 %	23.74 %	22.12 %	**23.05 %**	56.0/22.7
WDBC	4.16 %	4.33 %	1.83 %	**2.63 %**	1.77 %	2.93 %	23.3/18.5
WPBC	0.00 %	**0.00 %**	0.00 %	0.17 %	0.00 %	**0 %**	153.6/50.0

Pruning has been shown to be a useful trained ANN generalization improving step. The overall current chapter contributions are listed below.

- Developed pruning algorithm is presented, it is based on sensitivity measure with retraining, metric worsening threshold, and pocket memory, allowing pruning procedure to successfully escape local minimums.
- Conducted experiments have proven algorithm utility in simplifying the neural net structure and rising its generalization abilities.
- Input and hidden layer neurons pruning have shown lower classification errors, when compared to only hidden layers pruning.
- Experiments have shown that in general case nodes pruning is preferable over weights pruning as it requires a smaller amount of computations. The only exception would be a necessity to get the lowest possible error rate in which case weights pruning should be applied.

The current chapter accomplished task number two – the study of ANNs pruning approaches. The pruning approach was selected and based on the developed pruning algorithm.

3. DECISION TREE EXTRACTION FROM MULTILAYER PERCEPTRON

While frequently outperforming other types of classifiers artificial neural networks (ANN) black-box nature limit their usage. Therefore, knowledge extraction (KE) from trained ANN can help to uncover new knowledge, validate and productionalize or embed ANN-based classifiers.

3.1. Knowledge Extraction Overview

Description of various ANN types and architectures can be found in [12], [38], [56], [70], [92], [98], [120]. Due to the widespread usage of fully connected neuron layers, the current chapter concentrates on knowledge extraction from FFNNs. Non-linearity introduced by hidden layers is what makes FFNN flexible in modelling input data, but hard to understand classification decision. R. Setiono [96] was one of the first researchers who proposed to work not with weights of the neural network, but with neuron 'statistics' – neurons output values obtained over the training data set. As a preliminary KE step, it is proposed to prune neurons to minimize the number of neurons output values sets to be processed (see Chapter 2) – this potentially can minimize the number of 'rules' that will be extracted. The next stage involves neurons output values discretization to find bounds or, in the case of several neurons, regions from input space where all input vectors belonging to that region are classified as belonging to the same class. Later on, the developed algorithm (Fig. 3.1) can be applied to extract a classification decision tree from such discretized neurons.

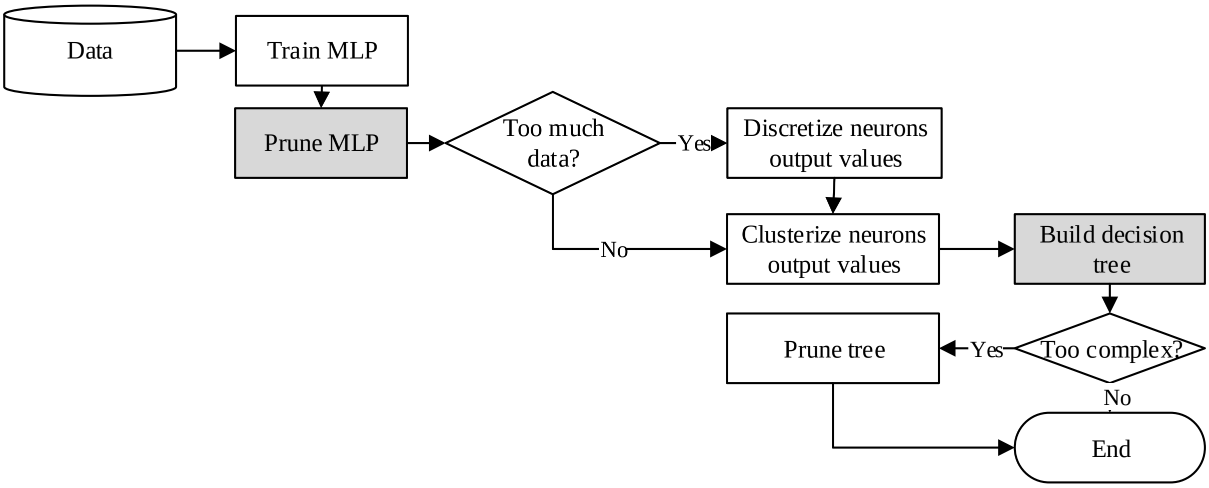


Fig. 3.1. The high-level knowledge extraction process.

The developed algorithm is decompositional as it is using intrinsic knowledge about neurons outputs in the neural network. Due to the knowledge extraction algorithm specifics a binary classification decision tree was chosen as a rules representation form. Its visualization allows for straightforward reasoning about the classification process. Decision trees can be directly, in a quazi-optimal way, mapped into If–Then rules as well.

Figure 3.1 provides a high-level overview of the knowledge extraction routine, along with required and optional pre-processing steps. Steps to which the current Thesis contributes to the development of new algorithms are highlighted.

In cases when the input data amount is prohibitively large, it is possible to: 1) work with the subset of the data; and 2) perform neurons output values discretization. After output values discretization step, output values clusterization needs to be performed that will further shrink possible neuron output values.

3.2. Knowledge Extraction Algorithm

Many of the KE algorithms described in the literature are extending the original NeuroRule algorithm [68]. NeuroRule and its derivatives have a very common workflow. The steps are neurons output signals non-linearity break-up via outputs clusterization (built up of quantization tables), afterwards all neurons starting from output layer are replaced with the sets of If–Then rules, finally all of these rules are merged and pruned to get final If–Then rules set. Instead of performing such rules merging, a solution was proposed to utilize only input neurons quantization tables for classification decision tree construction. In the proposed algorithm, to extract the decision tree from ANN, only quantization table borders are used (points at which class belongingness decision potentially changes) as a candidate split-points to estimate Information-Gain or GINI value on the full training dataset. This improvement lowers algorithm complexity and potentially amount of computations. To assure that the decision tree is accurately describing original ANN, the decision tree extraction dataset is generated, such that it contains points nearest to the quantized ANN classification boundary, see step 3 in Figure 3.2. A comparison of existing and proposed approaches can be seen in Figure 3.3.

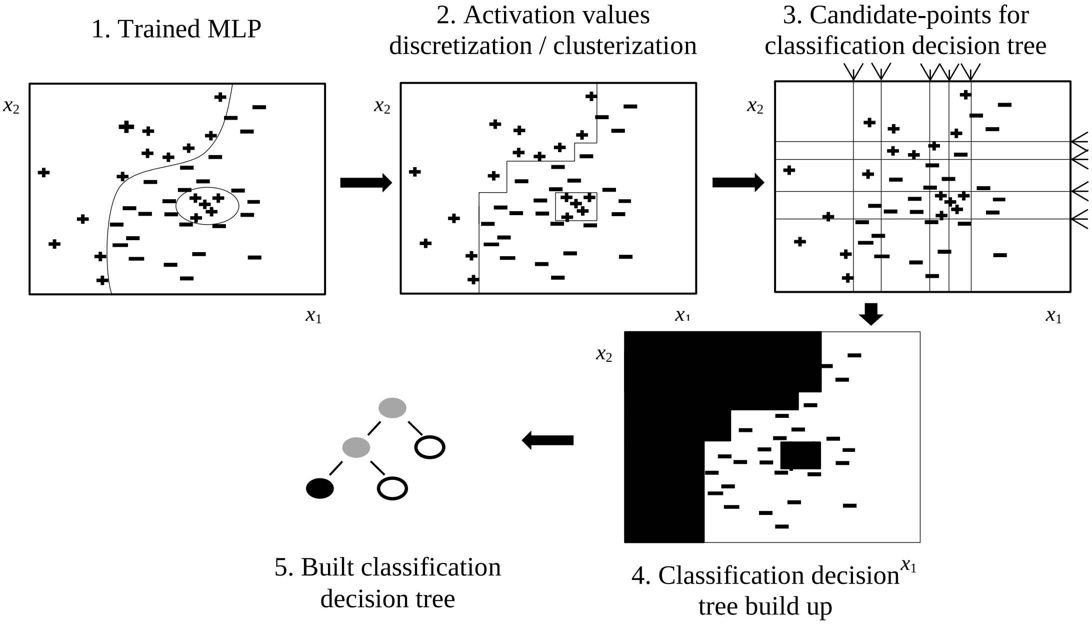

1. Trained MLP
2. Activation values discretization / clusterization
3. Candidate-points for classification decision tree
x_2
x_1
x_2
x_1
x_2
x_1
x_2
x_1
5. Built classification decision tree
4. Classification decision tree build up

Fig. 3.2. Decision tree built from clusters boundaries formed using input neurons outputs.

The main steps of the proposed algorithm for the decision tree extraction are (see steps 2-4 in Fig. 3.2):

1) pruned ANN classifier non-linearity is broken via neurons outputs clusterization;
2) candidate split-points are acquired from input layer neurons quantization tables;
3) a modified decision tree algorithm uses found candidate cluster boundaries from input layer neurons as a candidate split-points (while using full training dataset to calculate GINI or Information-Gain) to build a classification decision tree.

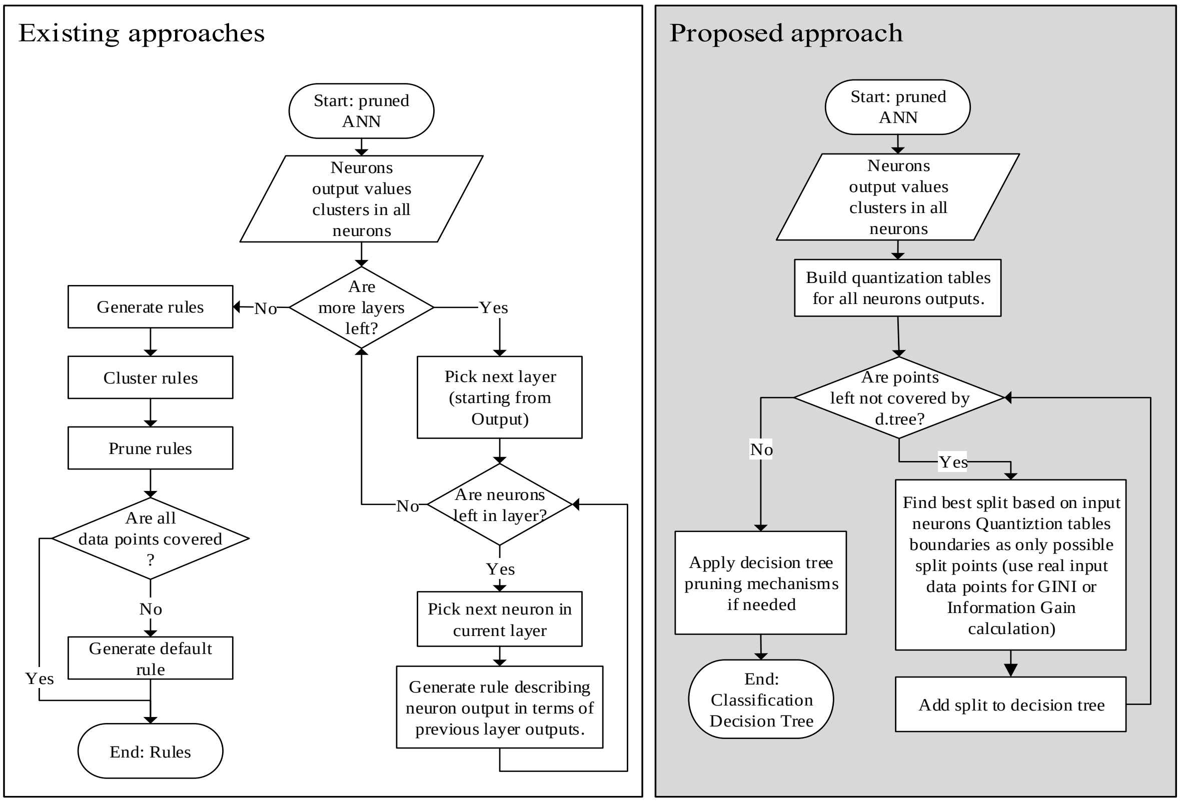

Existing approaches
Start: pruned ANN
Neurons output values clusters in all neurons
Are more layers left?
No
Yes
Generate rules
Cluster rules
Prune rules
Are all data points covered ?
No
Yes
Generate default rule
End: Rules
Pick next layer (starting from Output)
Are neurons left in layer?
No
Yes
Pick next neuron in current layer
Generate rule describing neuron output in terms of previous layer outputs.
Proposed approach
Start: pruned ANN
Neurons output values clusters in all neurons
Build quantization tables for all neurons outputs.
Are points left not covered by d.tree?
No
Yes
Apply decision tree pruning mechanisms if needed
End: Classification Decision Tree
Find best split based on input neurons Quantiztion tables boundaries as only possible split points (use real input data points for GINI or Information Gain calculation)
Add split to decision tree

Fig. 3.3. Existing and proposed approaches for rules extraction.

The described algorithms were implemented as an extension to the well-known deep learning package *nn* of *Torch7* library in *Lua* programming language [31]. Additionally, algorithms were reimplemented in Python programming language extending *PyTorch* deep learning library.

3.3. Developed Algorithm Validation

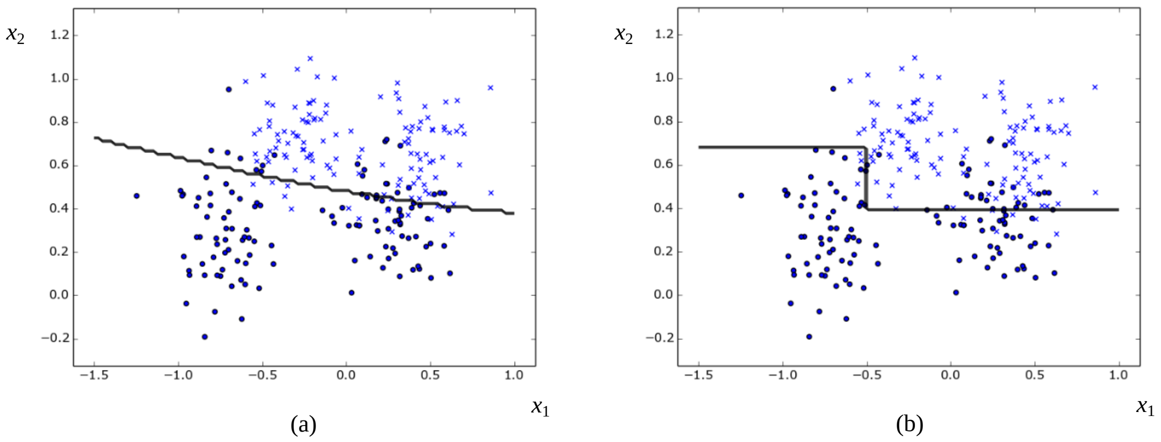
To validate the developed algorithm experiments have been performed on nine well-known UCI repository datasets. Table 3.1 holds the results of the experiments. For the C4.5 algorithm, results were taken from [2], [10], [17], [40], [74]. An example of the decision boundary for the pruned feedforward neural network is depicted in Figure 3.4(a), the decision boundary for the extracted binary decision tree can be seen in Figure 3.4(b). Average size decision tree, see Figure 3.5(a) and minimal and maximal, see Figure 3.5(b) extracted decision trees for Ripley and 3-class Iris datasets are presented in Figure 3.5.

Table 3.1

Accuracies and Leafs Counts for MLP, Pruned MLP, Extracted Tree, and C4.5

Dataset	**MLP train/test**	**Proposed methods**			**C4.5/J48 test**	**C4.5 leafs**
		Pruned MLP train/test	**Extracted tree train/test**	**Extracted tree leafs**$^{\min}_{\max}$		
Iris	0.9911/0.9667	0.9652/0.9667	0.9689/**0.9533**	4.2^{3}_{5}	0.9400	6
Pima diabetis	0.7319/0.7332	0.7253/0.7279	0.7433/**0.7423**	2^{2}_{2}	0.7210	14
Ionosphere	1.0000/0.9087	0.9552/0.9544	0.9546/**0.9059**	21.6^{12}_{26}	0.8971	17
Ripley	0.8560/0.8920	0.8640/0.8982	0.8631/**0.8946**	12.6^{4}_{20}	0.8941	6
Haberman	0.7509/0.7381	0.7542/0.7547	0.7567/**0.7446**	2^{2}_{2}	0.7190	n/a
Monks-1	1.0000/1.0000	1.0000/1.0000	1.0000/**1.0000**	8^{8}_{8}	1.0000	28
Monks-2	0.6923/0.6736	0.7160/0.6435	0.7041/0.5949	14^{14}_{14}	**0.6700**	1
Monks-3	0.9754/0.9259	0.9508/0.9722	0.9508/**1.0000**	6^{6}_{6}	0.9440	14
Parkinsons	0.8051/0.8001	0.8006/0.8099	0.8092/0.8102	2^{2}_{2}	**0.9261**	n/a

If there are a small number of cluster boundaries to be used as split points for decision tree building, then computational complexity is rather small. Experimental results show that extracted tree classification accuracy is directly related to a neural network accuracy, which is used as a knowledge source, as well as to the number of input neurons output values clusters boundaries that are used to build a tree. Authors of [74] have stated that they were able to get MLP accuracy of 0.9876, which is much higher than in presented experiments. This suggests that in the case of better trained MLP, the extracted tree would have higher classification accuracy. Of course, if comprehensibility and simplicity of the extracted decision tree are of higher importance, then more aggressive pruning can be applied and accuracy will be lower. But still, to get the best results it is important to have a trained ANN with the highest possible classification performance.



(a)

(b)

Fig. 3.4. Ripley test dataset split into two classes by (a) pruned neural network and (b) extracted decision tree.

Table 3.1 shows that the developed decision tree extraction algorithm is outperforming the C4.5 algorithm in the majority of cases in terms of classification accuracy. Cases when it shows lower performance are those were the original ANN has shown initial poor classification results. Rules-wise count both algorithms are on par, but having higher accuracy allows to further lower rules count, which gives the developed approach an edge over the C4.5 algorithm.

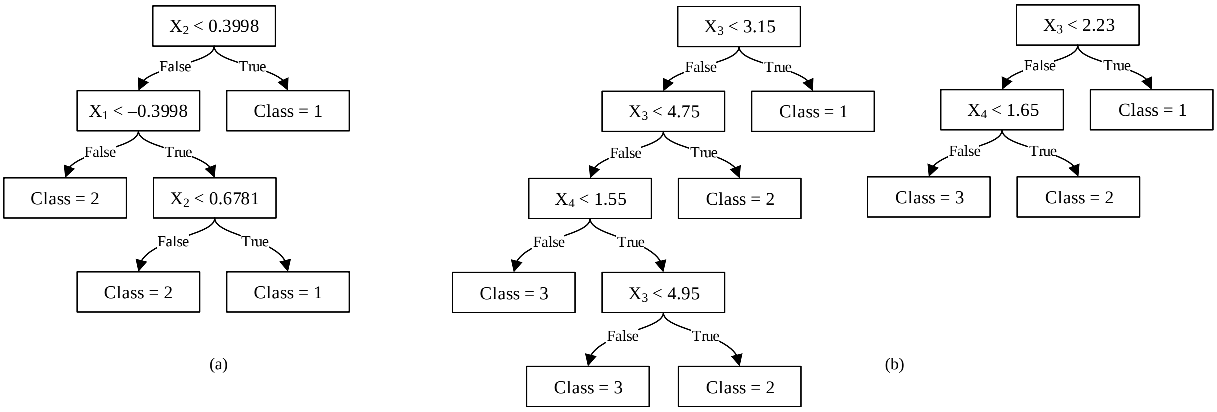


Fig. 3.5. Sample decision tree extracted for (a) Ripley data set and (b) decision trees of varying depth extracted for Iris data set.

Chapter 3 has reviewed knowledge extraction approaches in application to ANN; justified knowledge extraction from ANN in the form of a binary classification decision tree; justified the selection of Torch7 as a base for the deep learning framework for proposed NNKX [31] implementation; presented and experimentally validated the developed algorithm for classification decision tree extraction. The proposed algorithm has a lower classification error than C4.5 in all tested datasets, where a trained ANN had error lower than the C4.5 classifier. Experiments validated the developed approach. The extracted decision tree has high classification accuracy and low complexity, with that research task number three stated as accomplished.

4. OPTIMIZATION-BASED METHODS FOR RULE EXTRACTION

Chapter 3 introduced the decision tree extraction approach applicable to a fully connected trained feed-forward artificial neural network (ANN). But when it comes to training a classification model, it can be a case that the selected ANN will not show the best results. Therefore, Chapter 4 aims at the development of two alternative optimization-based approaches, one that allows the extraction of elliptical rules from radial basis function (RBF) neural networks and another approach allowing acquiring hyper-polytope classifier in input data space and approximating it using If–Then rules. Hyperpolytopes acquisition is a separate problem and two approaches to acquiring them are presented.

4.1. Elliptical Rules Extraction From RBF Neural Networks

Apart from fully connected ANNs with sigmoidal activation functions, RBF neural network (RBFNN) can be used as an alternative classifier. Elliptical rules are more expressive than If–Then rules. Therefore, optimization-based pedagogical approach allowing to extract elliptical rules from RBFNN was developed and evaluated [21], [27].

Optimization Problem

The extraction of elliptical rules from the trained RBFNN can be treated as a non-convex optimization problem of finding ellipsoids of maximum volume inscribed into the input space area defined by RFBNN classification decision boundary. Let us denote an ellipsoid as

$$\varepsilon = \{Bu + d \mid \|u\|_2 \leq 1\}, \tag{4.1}$$

where B is a symmetric positive definite matrix; u is a unit ball (set of points of distance one from a fixed central point); d is a vector representing ellipsoid center; $\|u\|_2$ denotes the Euclidean norm, i.e., $\|u\|_2 = (u^{\mathrm{T}}u)^{1/2}$.

Ellipsoid ε is a unit ball under affine transformation. In such a formulation, the ellipsoid volume is proportional to det$B$ [32]. Thus, the optimization problem can be posed as

$$\max \log(\det(B)),\ \text{s.t.}\,\text{RBFNN} \supseteq \varepsilon, \tag{4.2}$$

The described problem allows finding the first ellipsoid inscribed into the RBFNN decision boundary. A multi-start search will help in dealing with local optimums due to non-convexity of optimization problem. In most cases, it will be insufficient to represent RBFNN with a single ellipsoid; thus, iterative search for additional ellipsoids is required. To find other ellipsoids it is sufficient to look for the newly inscribed ellipsoid (potentially overlapping previously found ellipsoids) with maximum volume not covered by the already found ellipsoids:

$$\max(\varepsilon_{\text{vol}} - E_{\text{vol}}) - P,\, \text{s.t.}\,\text{RBFNN} \supseteq \varepsilon, \tag{4.3}$$

where ε_{vol} is the volume of newly found ellipsoid; E_{vol} is the volume of already existing (previously found) ellipsoids; and P is penalty term, see description below. Introduced penalty term P calculates the minimal distance (Eq. 4.4) between the candidate ellipsoid center and the border of a set formed by the intersection of all previously found ellipsoids.

$$P = \min(\mathrm{dist}(\varepsilon_{\mathrm{center}}, E_{\mathrm{surf}}) \colon E_{\mathrm{surf}} \in (E_1 \cup \ldots \cup E_N), \tag{4.4}$$

where $\varepsilon_{\mathrm{center}}$ is the center of newly found candidate ellipsoid; and E_{surf} is the surface formed by the intersection of already found ellipsoids. Introduction of P term ensures that on each iteration optimization objective will find a new ellipsoid, which will cover the largest possible portion of the volume not yet covered by existing ellipsoids.

Experiments and Results

Experiments have been conducted on a synthetic two-dimensional Ripley dataset (to aid visual analysis), which can be found in the UCI dataset repository [49]. The algorithm described in [37] was used in RBFNN initialization to construct several neural networks containing a variable number of neurons. In experiments only closed RBFNN defined classification boundaries were observed, which may be seen in figures. Looking at the algorithm, one can notice *maxEllipsoidsCount* variable. It was initialized with a number of neurons in the subject RBFNN, the only exception was a network with 9 neurons for which the maximum number of ellipsoids to be extracted had been set to 7. A number of neurons in RBFNN was chosen to be 2, 6, 7, and 9. Overall visual analysis confirms that the algorithm works as expected, while experimental results (see Table 4.1) show excellent performance of the extracted elliptical rules.

Table 4.1

RBFNN Accuracy, Extracted Ellipsoid Rules Accuracy, and Counts

# of neurons in RBFNN	RBFNN train accuracy	RBFNN test accuracy	Ellipsoid train accuracy$_{\text{std.dev.}}$	Ellipsoid test accuracy$_{\text{std.dev.}}$	Ellipsoids number mean$_{\max}^{\min}$
2 neurons	0.852	0.911	$0.8400_{0.000}$	$0.8870_{0.000}$	2_{2}^{2}
6 neurons	0.868	0.905	$0.8680_{0.002}$	$0.9032_{0.005}$	4.4_{5}^{4}
7 neurons	0.876	0.905	$0.8760_{0.000}$	$0.9031_{0.002}$	5.1_{7}^{4}
9 neurons	0.868	0.905	$0.8728_{0.005}$	$0.9039_{0.001}$	6.8_{7}^{5}

Experiment results show that the extracted ellipsoids have almost identical accuracy rates to original RBFNN while having an equal or smaller count of ellipsoids (in comparison to RBF neurons used in NN). Thus, the proposed algorithm works as expected. An important point to mention is computational complexity, as the algorithm uses RBFNN to check whether an ellipsoid fully lies within the RBFNN decision boundary. This is a computationally intensive operation, but it is parallelizable.

Summarized list of current sub-chapter contributions is as follows: optimization problem formulation, including a specific penalty; equipped objective function; and programmatic realization end experimental validation of the proposed algorithm. It was shown that a small number of found ellipsoids could perform classification with a small drop of classification accuracy. This proves the proposed approach to be feasible, especially for RBFNN with small input vector dimensionality.

4.2. Rules Extraction Using a Piece-Wise Approximation Algorithm

The current sub-chapter presents an application of optimization techniques for If–Then rules extraction from the piece-wise linear classifier. This direction was chosen because as it was already shown in [127], non-linear sigmoidal neurons decision boundary can be approximated by piece-wise linear functions. The developed algorithm is a generalization of the previously developed algorithm [80], which was developed to extract If–Then rules from linear support vector machine (SVM) classifier.

Overview of the Approach

To test If–Then rules extraction algorithm, a piece-wise polytope classifier was developed instead of piece-wise approximations of sigmoidal neurons outputs. The core idea of the developed piece-wise classifier is to build convex polytopes defined by hyperplanes around clusters of data points. Afterwards If–Then rules are extracted from such convex polytopes. Alternatively MSM-T classifier can be used as a source of polytopes [28]–[30], [79]. The developed algorithm is recursive, its main steps are:

1) acquisition of hyperpolytope classifier (Fig. 4.1(a));
2) allocation of best (by volume or by covered points count) If–Then rule (Fig. 4.1(b));
3) recursion start: splitting space into non-covered sub-spaces; finding the best If–Then rule in each of the found sub-spaces (Fig. 4.1));
4) checking recursion depth limit, if not reached, repeat step 3, otherwise stop (Fig. 4.1(b)).

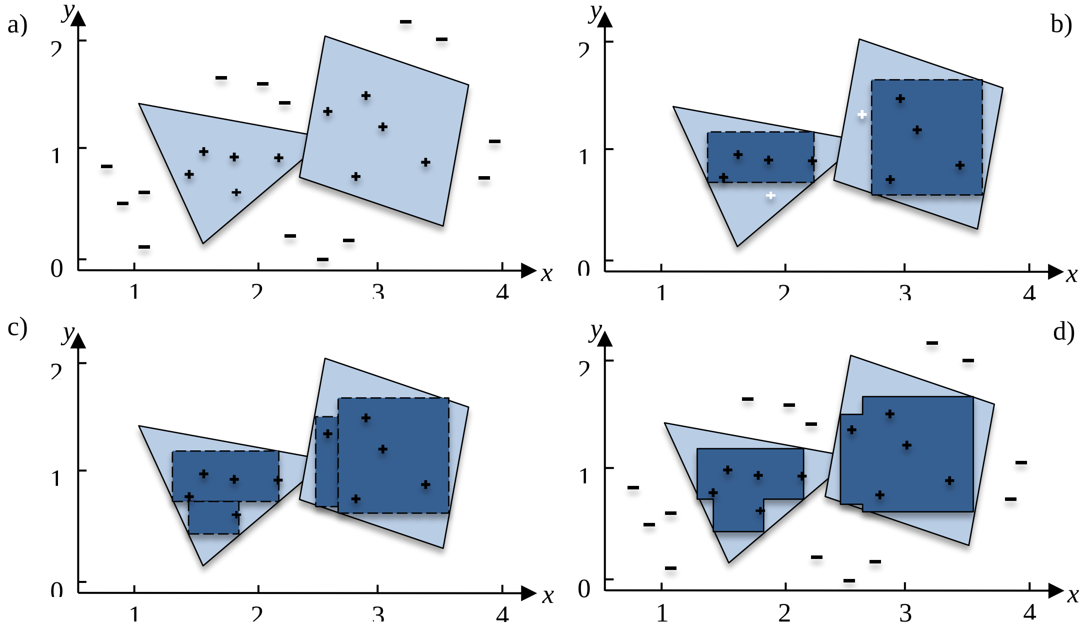


Fig. 4.1. If–Then Rule extraction algorithm iterations.

If–Then Rules Extraction Algorithm

The described algorithm follows the defined linear programming (LP) problem [28]–[30], [79] and defines two vertices, namely representing *lower* and *upper* bounds of the found hypercube. Having a single hypercube inscribed into the polytope can be insufficient in terms of classification fidelity. To overcome this undesirable result, the recursive search should be applied to search for additional hypercubes inscribed into the remaining regions of the polytope. This process could be repeated recursively so that ongoing search for the smaller uncovered regions will generate more and more rules that will asymptotically approximate the original polytope classifier with the desired level of classification fidelity. Such remaining regions of interest could be defined as follows:

$$I_i^l = \begin{cases} x \in R^n, \ \text{s.t.}, \\ l_j^* < x_j \leq u_j^* \forall 1 \leq j \leq i, \\ x_i \leq l_i^*, \end{cases} \qquad I_i^u = \begin{cases} x \in R^n, \ \text{s.t.}, \\ l_j^* \leq x_j < u_j^* \forall 1 \leq j \leq i, \\ x_i \geq u_i^*, \end{cases} \tag{4.5}$$

where I_i^l, I_i^u are polytope regions that are surrounding extracted rule for the i-th dimension; and l, u are upper and lower bounds of the currently processed hypercube (rule). Presented in Equation 4.5 rule inequalities are satisfied for the first $i-1$ dimensions of x, the inequality that relates to the i-th dimension is not satisfied, and the rest dimensions are free and should not be constrained. To support the recursive search, it is important to guarantee that new recursively inscribed hypercubes will not intersect with each other. Consider dimensions i, j with $j > i$. For each $x \in I_j$, we have $l_i^* < x_i < u_i^*$, and for each $x \in I_i$, we have $x_i \leq l_i^*$ or $x_i \geq u_i^*$. Hence, I_i are non-intersecting, and the rules that are acquired for each I_i differ in terms of approximated polytope region. It should be noted that polytopes could have intersections between each other, thus the extracted rules (hypercubes) could be intersecting. The optimization of extracted rules is not part of the current effort.

Experiments and Results

For the developed rule extraction approach verification and testing several public UCI datasets [49] were selected. The verification of the proposed method is covered in [28]–[30], [79]. Datasets for the experiments were selected based on popularity criteria.

Before the actual rule extraction algorithm can take place, polytopes should be created. To acquire polytopes from which If–Then rules will be extracted it is possible to use the MSM-T classifier, which is as a classification decision tree with optimal splits that are not parallel to axes. Another option is to use a proposed piece-wise linear classifier, see [79] for details of the proposed method. For datasets that were not originally separated into validation and training sets 10-fold cross-validation was performed, and averaged classification accuracy was collected. In the case of Monks dataset training and validation, datasets were already provided and 10 experiments were conducted.

It can be seen in Table 4.2 that all datasets, except “Balance-Scale”, are not so nicely separable using Linear SVM. On the other hand, an approximation of a nonlinear decision surface gives a necessary boost of the classification accuracy for the polytope classifier and for extracted rules. Here C4.5 fails to perform good classification. Multi-Surface Method Tree (MSM-T) method falls behind SVM methods and Polytopes. The rules extracted from MSM-T show high classification error. Empirically it was found that the increase in recursion depth for rules extraction from MSM-T helps to lower the classification error.

Table 4.2

Classification Accuracy, %

Classifier	Monks-1	Monks-2	NDCC	Balance
SVMlinear	65.509	67.130	73.333	93.730
SVMrbf	86.806	**80.556**	**95.000**	**98.082**
MSM-T	83.565	79.630	88.667	87.526
Rules (MSM-T)	69.444	54.861	75.333	68.524
C4.5	75.690	65.050	74.000	70.780
Polytopes	**99.537**	80.324	93.667	97.913
Rules (polytopes)	96.296	74.537	89.738	96.857

The rows highlighted in grey are the results shown by developed classifiers. Bold highlights the best overall classification result on test set across all datasets (columns). Green, yellow and red are highlighting explainable classifiers accuracies. Here one can see that the rules extracted from polytope classifier have higher accuracy than rules extracted from MSM-T. Overall SVM with RBF kernel outperforms other classifiers. Among all classifiers, only three classifiers (C4.5, Rules (MSM-T) and Rules (polytopes)) are explainable. Among them, the rules extracted from hyper-polytopes (lower row) have the highest accuracies. This proves that the chosen approach can be successfully applied to extract precise rules.

The current chapter presents two approaches developed in the scope of optimization-based knowledge extraction. These approaches complement and serve as an alternative methods of knowledge extraction in cases when the decision tree extraction from trained ANN has shown poor results. Also, in cases when ANN to be described by the set of the rule is RBFNN and we are willing to lower the number of rules – elliptical rules are a better option. In case when one has hyperpolytopes to be described with If–Then rules, the developed approach provides yet another alternative to classification decision tree extraction.

Experimental validation has shown that If–Then rules can be extracted from hyper-polytopes using the developed algorithm as a convex-optimization problem solution. It was shown that extracted If–Then rules have high accuracy, thus can be used to describe original classifier and serve as yet another tool in machine learning practitioner toolbox. Also, a non-convex optimization problem was posed, and a new algorithm developed that supports the extraction of Elliptical rules. The above said allows concluding that the research task related to development and assessment of optimization-based methods for If–Then and elliptical rule extraction from trained FFNN and RBFNN is accomplished.

Research task number four, related to the development and assessment of optimization-based methods for If–Then and elliptical rules extraction from trained FFNN and RBFNN is accomplished.

5. KNOWLEDGE EXTRACTION METHODOLOGY

Based on accomplished work and acquired results, a methodology for rules extraction is developed and experimentally validated in the current chapter. Unified workflow for knowledge representation and model selection, knowledge extraction, assessment, and refinement is presented. Review of knowledge representation and extraction from artificial neural networks and other types of classifiers covered in Chapter 1 serves as a basis for the methodology, which presents a pruning algorithm, see Chapter 2, a novel algorithm for binary classification decision tree extraction from fully connected multilayer ANN, covered in Chapter 3. Finally, optimization-based method for rules extraction using convex optimization problem is covered in Chapter 4. The same chapter presents a novel developed (non-convex optimization) method for elliptical rules extraction from RBFNN classifier. The proposed workflow [18]–[20] is experimentally validated and guidelines on how to acquire simple or precise rules are given.

5.1. Methodology Development

Methodology developed as part of the dissertation divides knowledge extraction (KE) process into four main stages.

1. Knowledge representation schema and classifier type selection.
2. Classifier training and preparation for KE steps.
3. Knowledge extraction.
4. Extracted knowledge assessment and refinement.

The first step assumes knowledge schema selection. Elliptical rules are more expressive than decision tree or If–Then rules, but less comprehensible. So, if the overall goal is to understand how classification is performed elliptical rules might be a suboptimal choice. On the other side decision tree can be extracted only from ANN, hence if the best performing classifier is piece-wise hyperpolytopes then If–Then rules will have to be used as knowledge representation.

The second and third steps are straightforward. In case the classifier is already pre-trained KE can be performed right away. In the case of ANN pruning can be applied, which can be used to influence on precision and number of extracted rules.

The fourth step is knowledge assessment and refinement. If extracted rules are too complex, several possibilities to mitigate that are listed below.

- Rules can be merged and pruned.
- A greater degree of pruning should be applied on ANN, or lower amount of ellipsoids or smaller depth of If–Then rules recursion should be set.
- The classifier could be retrained to have altered training parameters and hyperparameters and KE steps should be re-run.
- As an alternative, if dataset size allows it, smaller sub-space region, were used classifier performance is poor, can be used to train a separate classifier to run KE on it.

All developed algorithms are combined into a unified methodology, see Figure 5.1.

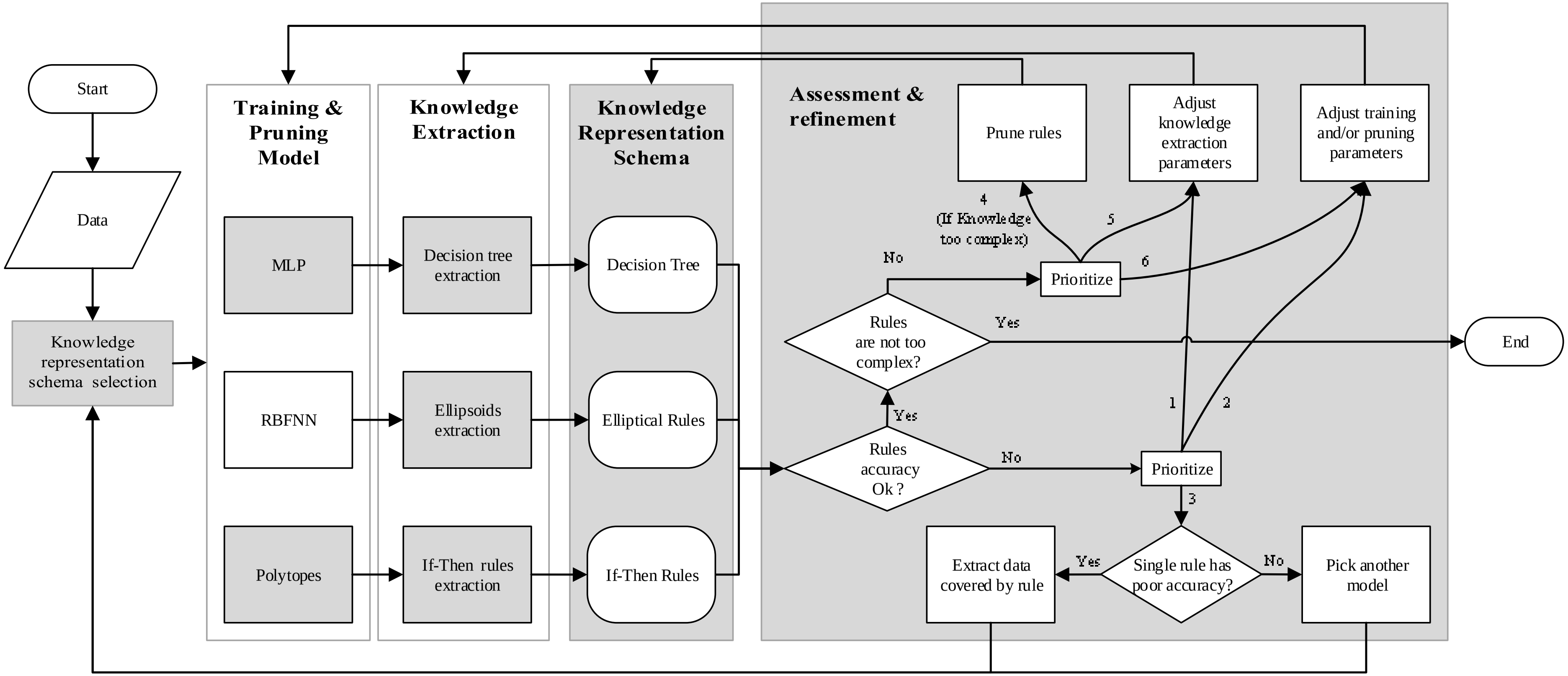


Fig. 5.1. Methodology for knowledge extraction. Contribution of the Thesis is highlighted in grey.

5.2. Precise vs. Comprehensive Rules

To validate the proposed workflow and assess guidelines for acquiring precise or simple rules, an experiments plan has been developed and executed on medium and small sized datasets. To overcome large computational requirements needed to process real-life medium-sized (~50 000 records) Adult Census dataset (taken from UCI repository), two modifications were incorporated into the KE algorithm:

- neurons output discretization via outputs rounding to *n* digits for all neurons become a mandatory step, disregarding possible performance degradation;
- neurons outputs clusterization was performed on a data subset (in experiments, it was 15 % of the training set).

The experiments goal is to prove that the parameter controlling allowed performance metric degradation during neurons outputs clusterization phase allows controlling the extracted decision trees complexity and classification performance. The second question was to understand how pruning, as a preceding step, influences the extracted decision tree complexity and performance.

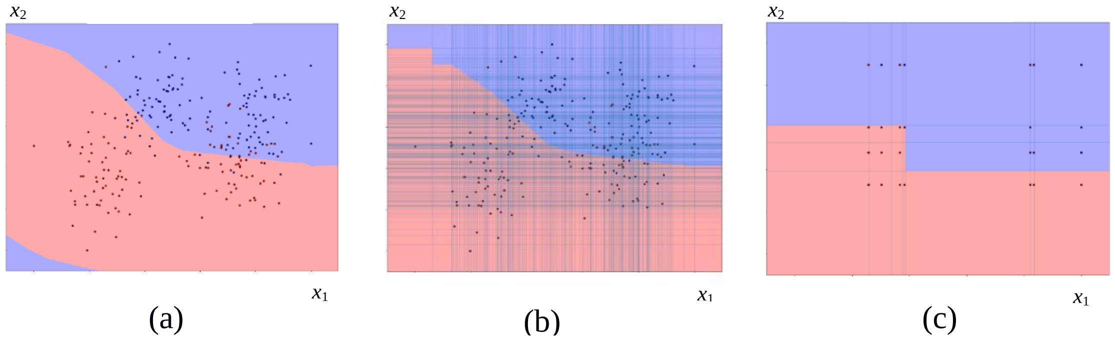
Results gathered in Table 5.1 hold means over ten experiments for each dataset and show that aggressive pruning coupled with large performance degradation threshold for neurons outputs clustering results in lower classification performance and smaller decision trees. Light pruning and small threshold give larger, but more precise decision trees.

Table 5.1

Knowledge Extraction Parameters Influence on Extracted Rules Complexity and Accuracy

ANN pruning level	**Neurons output values clusterization**	**Characteristics**	**Datasets**			
			Adult		**Ripley**	
			Train	**Test**	**Train**	**Test**
Aggressive pruning	ANN		82.27	82.15	83.84	87.92
	Low perf. degradation clustering threshold	Discretized ANN Accuracy, %	82.09	81.95	83.84	87.89
		Extracted tree accuracy, %	79.74	79.81	83.8	88.0
		Extracted tree rules count / depth	212.7/16.2		55.6/28.7	
	Medium perf. degradation clustering threshold	Discretized ANN accuracy, %	80.89	80.99	83.28	87.76
		Extracted tree accuracy, %	79.80	79.74	83.28	87.76
		Extracted tree rules count / depth	20.1/7.7		16.8/12.3	
Minimal pruning	ANN		83.41	83.22	84.44	89.52
	Low perf. degradation clustering threshold	Discretized ANN accuracy, %	83.39	83.21	84.44	89.62
		Extracted tree accuracy, %	82.12	81.95	84.44	89.52
		Extracted tree rules count / depth	1562/21.8		137/91.6	
	Medium perf. degradation clustering threshold	Discretized ANN accuracy, %	82.27	82.00	83.4	88.4
		Extracted tree accuracy, %	79.25	78.87	83.4	88.4
		Extracted tree rules count / depth	161.1/12.5		11.3/8.1	

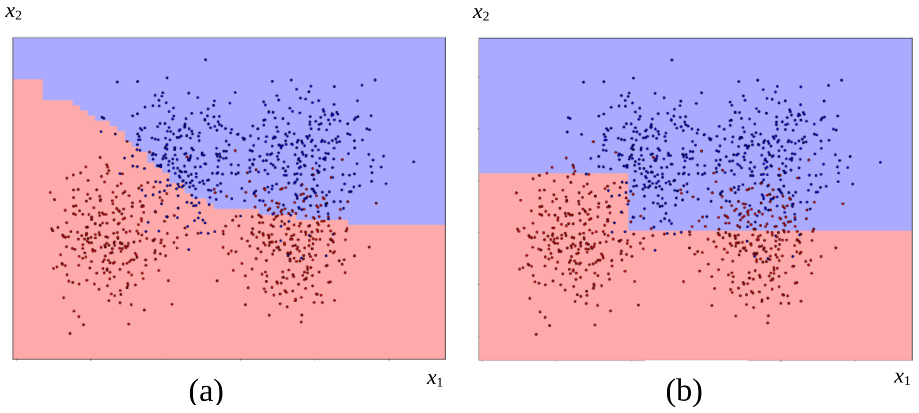
Experiments on the Ripley dataset help to understand how the clusterization threshold parameter influences the extracted decision tree. As a starting point, Figure 5.2(a) displays how the initial classification boundary produced by trained and slightly pruned ANN looks like. The result of discretization and clusterization using small (Fig. 5.2(b)) and large (Fig. 5.2(c)) allowable performance degradation threshold is seen in Figure 5.2. Vertical and horizontal lines show clusters boundaries.



(a) (b) (c)

Fig. 5.2. Classification boundaries for the Ripley dataset.

A small number of points in Figure 5.2(c) is due to the utilization of quantization tables that are replacing neurons output values. Figure 5.3 shows the classification boundaries of decision trees extracted from (quantized) ANN using neurons outputs clusterization.



(a) (b)

Fig. 5.3. Decision tree classification boundary of an ANN with (a) small and (b) large clustering parameter controlling performance degradation.

Experiments have shown that the primary tool for controlling the extracted decision tree complexity is the clusterization phase performance degradation threshold. Next auxiliary parameter belongs to discretization – how big rounding should be, to acquire a smaller count of neurons outputs and lower computational costs during the clusterization phase (at the cost of some classification performance degradation). Lastly, pruning itself can control the rules complexity and classification performance, but the pruning effect has smaller importance on extracted tree complexity.

In regards to accuracy, pruning plays an important role as a regularizer preventing overfitting. In any case, when non-agressive pruning is applied, the extracted tree will have classification performance similar to ANN it is being extracted from.

The current chapter summarizes the conducted work and presents a unified workflow with recomendations for KE. This workflow and recomendations underline the research work accomplished in previous chapters and provides guidelines for knowledge extraction with experimental validation results. Guidelines are provided for the selection of classifiers and corresponding knowledge extraction algorithms, taking into account dataset characteristics. Workflow for dealing with overly complex or too simple extracted knowledge is presented and experimentally validated. The contribution of current work can be summarized as follows:

- proposed methodology – general workflow of knowledge extraction from trained ANN or hyper-polytope classifier;
- Recommendations formulated within the scope of methodology on classification model selection based on dataset characteristics;
- as part of the methodology, based on experimental validation proposed recommendations on pruning and knowledge extraction parameters selection, described parameters influence on extracted knowledge complexity and performance;
- defined assessment procedure suggesting how knowledge should be assessed and what corrective actions can be performed to fix problems (if any are found).

Classification decision tree extraction is the most general way of acquiring knowledge from a neural network. However, according to the “No Free Lunch” theorem, there is no single method, which will be equally good for all datasets. Thus, alternative algorithms for extraction of elliptical and If–Then rules were developed and proved to be usable. All listed improvements have become parts of the overall knowledge extraction workflow. The work presented in this chapter accomplishes task number five stated as a research task.

RESULTS AND CONCLUSIONS

The Doctoral Thesis is devoted to knowledge extraction from trained artificial neural networks (ANN). Within this work, an analysis of existing approaches to knowledge representation, ANN pruning, and knowledge extraction was performed. As a result of this analysis, a methodology for knowledge extraction was developed. This methodology lists typical knowledge representation schemes, and provides guidelines for selecting best knowledge representation. In the case when knowledge is extracted from ANN, due to lack of ready to use algorithms and low performance of available implementations, new pruning approach was developed. An algorithm for classification decision tree extraction from ANN was developed. To cover more use-cases, alternative optimization-based knowledge extraction approaches were developed. These approaches allow extraction of If–Then rules from the classifier described by hyperpolytopes and Elliptical rules from Radial-Basis function neural network (RBFNN). All developed approaches are united into single workflow along with recomendations in regards to choosing specific workflow steps. The decision tree classifier is supplied with experimentally proven recommendations allowing user to get either complex and precise or more comprehensible and less precise decision trees.

The aim of the Doctoral Thesis is to develop algorithms for ANN pruning and knowledge extraction from trained ANN and unify them into knowledge extraction methodology. Thesis accomplishments are as follows.

1. A review and analysis of existing scientific literature covering theory and algorithms for knowledge representation and extraction is performed. As a result, the research discovered pros and cons of existing approaches, which allowed to define requirements for knowledge extraction workflow.
2. Artificial neural networks pruning algorithms are reviewed, and new ANN pruning algorithm based on sensitivity measure with retraining and pocket memory is developed and evaluated. Recommendations for choosing weights vs. nodes pruning are developed. The algorithm ability to overcome local minimums is proven experimentally.
3. A new decompositional approach for classification decision tree extraction from trained multilayer perceptron is developed. The algorithm extends Torch7 based *nn* deep learning package with an additional neural network.
4. Optimization based pedagogical approaches for oblique If–Then and elliptical rules extraction from a set of convex hyper-polytopes and RBF neural network are developed and evaluated.
5. Generalized methodology for knowledge extraction is developed and evaluated. The methodology includes suggestions on the model selection (MLP, RBFNN, and convex polytopes), which influences knowledge representation and extraction approach. Methodology contains workflow, which guides knowledge extraction, assessment, and refinement depending on the selected classification model. The experimental validation of methodology testifies the conclusions.

The performance of all developed algorithms was evaluated and analysed to prove the proposed hypotheses. Based on the conducted research, several conclusions can be made.

- The developed pruning algorithm based on sensitivity analysis successfully escapes from local minimums and allows to control classification error rise. In the scope of sensitivity-based pruning in some cases, weights pruning can produce results better than neurons pruning. Although neurons pruning is more welcome as it is a less computationally intensive method and in general produces results on par with weights pruning, these findings prove **the first hypothesis.**
- Usage of retraining and memory-pocket trick are simple yet effective algorithmic improvements that, when used with sensitivity-based pruning algorithm, produce good pruning results – these findings contribute to **the first hypothesis** as well**.**
- In the scope of decision tree extraction from MLP, usage of input layer neurons output values classification decision boundaries (acquired via neuron output values clusterization) instead of replacement of all neurons with rules produces a good classification decision tree. Such method is simpler in terms of computational resources in comparison to the description of all neurons via sets of rules with subsequent rule clustering, merging and pruning (to get required rules for the input layer). Additionally, the developed approach for classification decision tree extraction allows to control the extracted tree complexity and classification accuracy – this proves **the second hypothesis.**
- Based on experiments results involving the developed approach, a conclusion can be made that optimization-based approach can be used for If–Then rules extraction from convex polytopes. On non-HPC hardware, this method is applicable to datasets with less than eleven attributes. Hence, this method is usable on subsets of input datasets as an alternative method in case the extraction of classification decision tree from MLP produces poor classification decision tree for a specific sub-region. The experiments prove that the extracted If–Then rules effectively approximate the input space regions bounded by hyper-polytopes – this proves **the third hypothesis.**
- Based on experiments involving the developed algorithm, a conclusion can be made that the optimization-based approach can be used for Elliptical rule extraction from RBFNN for datasets with less than four dimensions (on non-HPC hardware). This method is applicable as a way to replace large sub-trees in a decision tree with more expressive elliptical rules. The experiments prove that the extracted elliptical rules effectively approximate RBFNN and achieve similar classification accuracy – this proves **the fourth hypothesis.**

All posed theoretical questions are experimentally proven via proposed and developed approaches and methodology. As a result, in the scope of Thesis rules extraction methodology is developed and experimentally evaluated. The methodology allows performing a selection of knowledge representation schema and classification method, knowledge extraction, its assessment, and refinement. Further research directions can include research of ways to introduce reproducibility into ANN pruning and rules extraction, as well as neurons output values clusterization speedup.

BIBLIOGRAPHY

1. Abadi, M., Agarwal, A., Barham, P., et al. *TensorFlow: Large-scale Machine Learning on Heterogeneous Systems* [online]. 2015 [viewed 1 June 2019]. Available from: https://www.tensorflow.org/extras/tensorflow-whitepaper2015.pdf.
2. Abdelhalim, A., Traore, I., Sayed, B. RBDT-1: A New Rule-Based Decision Tree Generation Technique. In: G. Governatori, J. Hall, A. Paschke, eds. *Rule Interchange and Applications. RuleML 2009*. Berlin, Heidelberg: Springer, 2009, pp. 108–121.
3. Alshahrani, M., Soufan, O., Magana-Mora, A., Bajic, V. B. DANNP: An Efficient Artificial Neural Network Pruning Tool. *PeerJ Computer Science*. 2017. Available from: https://doi.org/10.7717/peerj-cs.137.
4. Anderson, J. A. *An Introduction to Neural Networks*. MIT Press, 1995. 672 p.
5. Augasta, M. G., Kathirvalavakumar, T. A Novel Pruning Algorithm for Optimizing Feedforward Neural Network of Classification Problems. *Neural Processing Letters*. 2011, vol. 34, no. 3, pp. 241–258.
6. Augasta, M. G., Kathirvalavakumar, T. Pruning Algorithms of Neural Networks – A Comparative Study. *Central Europ. J. of Computer Science*. 2013, vol. 3, no. 3, pp. 105–115.
7. Augasta, M. G., Kathirvalavakumar, T. Reverse Engineering the Neural Networks for Rule Extraction in Classification Problems. *Neural Processing Letters.* 2012, vol. 35, no. 2, pp. 131–150.
8. Babaeizadeh, M., Smaragdis, P., Campbell, R. H. *NoiseOut: A Simple Way to Prune Neural Networks* [online]. 2016 [viewed 1 June 2019]. Available from: https://arxiv.org/pdf/1611.06211.pdf.
9. Baesens, B., Gestel, V., Viaene, T. S., et al. Benchmarking State-of-the-art Classification Algorithms for Credit Scoring. *Journal of the Operational Research Society*. 2003, vol. 54, no. 6, pp. 627–635.
10. Belacel, N., Al-Obeidat, F. Learning PROAFTN with a Comparative Study with Decision Trees. In: C. J. Butz and P. Lingras, eds. *Advances in Artificial Intelligence: 24th Canadian Conference on Artificial Intelligence, Canadian AI 2011, St.John's, Canada, 25*–27 *May 2011, Proceedings*. Berlin: Springer, 2011, pp. 56–61. (Lecture Notes in Artificial Intelligence. Vol. 6657).
11. Barakat, N., Diederich, J. Eclectic Rule-Extraction from Support Vector Machines. *Intern. J. of Computational Intelligence.* 2005, vol. 2, no. 1, pp. 59–62.
12. Bengio, Y. Learning Deep Architectures for AI. *Foundations and Trends® in Machine Learning.* 2009, vol. 2, no. 1, pp. 1–127.
13. Beltramelli, T. *pix2code: Generating Code from a Graphical User Interface Screenshot* [online]. Copenhagen: UIzard Technologie, 2017 [viewed 1 June 2019]. Available from: https://uizard.io/research#pix2code.

14. Bilbao, I., Bilbao J. Overfitting Problem and the Over-training in the Era of Data: Particularly for Artificial Neural Networks. In: *The 8th International Conference on Intelligent Computing and Information Systems (ICICIS), Cairo, Egypt. 2017, 5–7 December 2017*. IEEE, 2017. Available from: doi:10.1109/INTELCIS.2017.8260032.
15. Bishop, C. M. B6. Neural Networks: A Pattern Recognition Perspective. In: E. Fiesler, R. Beale, eds. *Handbook of Neural Computation*. Oxford: Oxford University Press. 1997, pp. 190–212.
16. Bodyanskiy, Ye. V., Rudenko, O. G. *Artificial Neural Networks: Architectures, Learning, Applications*. Kharkiv: TELETEH, 2004. 372 p. (in Russian).
17. Bondarenko, A., Aleksejeva, L., Jumutcs, V., Borisovs, A. Classification Tree Extraction from Trained Artificial Neural Networks. *Procedia Computer Science,* 2017, vol. 104, pp. 556–563.
18. Bondarenko, A., Aleksejeva, L. Methodology for Knowledge Extraction from Trained Artificial Neural Networks. *Information Technology and Management Science*. 2018, vol. 21, pp. 6–14.
19. Bondarenko, A., Aleksejeva, L. Workflow for Knowledge Extraction from Neural Network Classifiers. In: *59th Intern. Scientific Conf. on Information Technology and Management Science of Riga Technical University (ITMS), 2018*. Available from: doi:10.1109/ITMS.2018.8552964.
20. Bondarenko, A., Aleksejeva, L. Controlling Complexity and Accuracy of Classification Decision Tree Extracted from Trained Artificial Neural Network. In: *60th International Scientific Conference,* 10–11 October 2019 Available from doi:10.1109/ITMS47855.2019.8940739
21. Bondarenko, A., Borisov, A. The Extraction of Elliptical Rules from the Trained Radial Basis Function Neural Network. *Information Technology and Management Science*. 2012, vol. 15, pp. 161–165. Available from: doi:10.2478/v10313-012-0027-2.
22. Bondarenko, A., Borisov, A. Decompositional Rules Extraction Methods from Neural Networks. In: *Proceedings of the 16th International Conference on Soft Computing MENDEL'10, Brno, Czech Republic, 23–25 June 2010*. Brno: University of Technology, 2010, pp. 256–262.
23. Bondarenko, A., Borisov, A. Research of Artificial Neural Networks Abilities in Printed Words Recognition. *Information Technology and Management Science*. 2010, vol. 44, issue 5, pp. 124–129.
24. Bondarenko, A., Borisov, A. Research on the classification ability of deep belief networks on small and medium datasets. *Scientific Journal of Riga Technical University, Information Technology and Management Science*. 2013, vol. 16, pp. 60–65.
25. Bondarenko, A., Borisovs, A., Aleksejeva, L. Neurons vs Weights Pruning in Artificial Neural Networks. In: *Environment. Technology. Resources: Proceedings of the 10th International Scientific and Practical Conference, Rezekne, Latvia, 18–20 June 2015*. Vol. 3. Rezekne: Rezekne Higher Education Institution, 2015, pp. 22–28.

26. Bondarenko, A., Borisovs, A. Artificial Neural Network Generalization and Simplification via Pruning. *Information Technology and Management Science*. 2014, vol. 17, pp. 132–137.
27. Bondarenko, A., Borisovs, A. Elliptical Rule Extraction from a Trained Radial Basis Function Neural Network. In: *The 6th International Conference "Applied Information and Communication Technology", Jelgava, Latvia, 25–26 April 2013*. Jelgava: LUA Faculty of Information Technology, 2013 (CD-ROM).
28. Bondarenko, A., Borisovs, A. Knowledge Extraction from Piecewise-Linear Approximation of Multi-Surface Classifier. In: *International Conference "Information Intelligent Systems", Kharkov, Ukraine, 17–19 April, 2012*. Vol. 6, pp. 5–6.
29. Bondarenko, A., Jumutc, V. Extraction of Interpretable Rules from Piecewise-Linear Approximation of a Nonlinear Classifier Using Clustering-Based Decomposition. In: *Proceedings of the 10th WSEAS International Conference on Artificial Intelligence, Knowledge Engineering and Data Bases (AIKED'11), Cambridge, United Kingdom, 22–22 February 2011*. Cambridge: 2011, pp. 145–149.
30. Bondarenko, A., Zmanovska, T., Borisovs, A. Piece-Wise Classifier Application to RBF Neural Network Rules Extraction. In: *17th International Conference on Soft Computing (MENDEL'11), Brno, Czech Republic, 15–17 June 2011*. Brno: Brno University of Technology, 2011, pp. 170–176.
31. Bondarenko, A. *NNKX – Neural Networks Knowledge eXtraction* [online]. Publisher: bitbucket.org, 2018 [viewed 1 June 2019]. Available from: https://bitbucket.org/bondtnt/nnkx
32. Boyd, S., Vandenberghe, L. *Convex Optimization.* Cambridge: Cambridge University Press, 2004. 727 p.
33. Brockman, J. *Thinking: The New Science of Decision-Making, Problem-Solving, and Prediction*. Harper Perennial, 2013. 432 p. (Best of Edge Series).
34. Buhmann, M. D. *Radial Basis Functions: Theory and Implementations*. Cambridge University, 2003. 272 p.
35. Castro, J. L., Mantas, C. J., Benítez, J. M. Interpretation of Artificial Neural Networks by Means of Fuzzy Rules. *IEEE Transactions on Neural Networks.* 2002, vol. 13, no. 1, pp. 101–116.
36. Chaudhary, V., Ahlawat, A. K., Bhatia, R. S. Growing Neural Networks using Soft Competitive Learning. *International Journal of Computer Applications.* 2011, vol. 21, no. 3, pp. 1–6. Available from: doi:10.5120/2495-3372.
37. Chen, S., Hong, X., Luk, B.L., et al. Construction of Tunable Radial Basis Function Networks using Orthogonal Forward Selection. *IEEE Trans. Systems, Man, and Cybernetics, Part B: Cybernetics*. 2009, vol. 39, no. 2, pp. 457–466.
38. Chung, J., Gulcehre, C., Cho, K. H., et al. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. *NIPS 2014 Workshop on Deep Learning,* December 2014. Available from: https://arxiv.org/abs/1412.3555.

39. Craven, M. W. *Extracting comprehensible models from trained neural networks*. Ph.D. thesis. Madison: University of Wisconsin–Madison, 1996. 199 p. Available from: https://www.biostat.wisc.edu/~craven/papers/thesis.pdf.

40. De Backer, M., Haesen, R., Martens, D., et al. A Stigmergy Based Approach to Data Mining. In: S. Zhang, R. Jarvis, eds. *AI 2005: Advances in Artificial Intelligence: 18th Australian Joint Conference on Artificial Intelligence, Sydney, Australia, 5–9 December 2005*. Berlin Heidelberg Springer-Verlag 2005, p. 975–978. (Lecture Notes in Artificial Intelligence. Vol. 3809).

41. *Deeplearning4j: Open-source, Distributed Deep Learning for the JVM* [online]. Deeplearning4j Team, [viewed 1 June 2019]. Available from: https://deeplearning4j.org/.

42. Deng, L. Deep Learning: From Speech Recognition to Language and Multimodal Processing. *APSIPA Transactions on Signal and Information Processing*. 2016, vol. 5, e1, pp. 1–15. Available from: doi:10.1017/ATSIP.2015.22.

43. Denil, M., Bazzani, L., Larochelle, L., et.al. 2011. Learning where to Attend with Deep Architectures for Image Tracking. CoRR. [Online]. Available from: https://arxiv.org/abs/1109.3737 [Accessed: 1 June 2019].

44. Dienes, Z., Perner, J. A Theory of Implicit and Explicit Knowledge. *Behavioral and Brain Sciences.* 1999, vol. 22, pp. 735–808.

45. Domingos, P. A Unified Bias-Variance Decomposition and Its Applications. In: *Proceedings of the 17th International Conference on Machine Learning, ICML'2000, Stanford, CA, USA, 29 June – 2 July 2000.* San Francisco: Morgan Kaufmann, 2000, pp. 231–238.

46. Dong, X., Chen, S., Pan, S. J. *Learning to Prune Deep Neural Networks via Layer-wise Optimal Brain Surgeon* [online]. 2017 [viewed 1 June 2019]. Available from: https://arxiv.org/pdf/1705.07565.pdf.

47. Dua, D. and Graff, C. *UCI Machine Learning Repository* [online]. Irvine, CA: University of California, School of Information and Computer Science, 2019 [viewed 10 June 2019]. Available from: http://archive.ics.uci.edu/ml.

48. Erhan, D., Bengio, Y., Courville, A., et al. Why Does Unsupervised Pre-training Help Deep Learning? *Journal of Machine Learning Research*. 2010, vol. 11, pp. 625–660.

49. Fiesler, E., Cios, K. C1.7 Supervised ontogenic networks. In: E. Fiesler, R. Beale, eds. *Handbook of Neural Computation.* Oxford University Press, 1997, pp. 320–337.

50. Fukushima, K. Cognitron: A Self-organizing Multilayered Neural Network. *Biological Cybernetics*. 1975, vol. 20, no. 3–4, pp. 121–136. Available from: doi:10.1007/BF00342633.

51. Fukushima, K. Neocognitron: A Self-organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position. *Biological Cybernetics*. 1980, vol. 36, no. 4, pp. 193–202. Available from: doi:10.1007/BF00344251.

52. Fung, G., Sandilya, S., Bharat Rao, R. Rule Extraction from Linear Support Vector Machines. In: *KDD'05 Proceedings of 11th ACM SIGKDD International Conference on Knowledge Discovery in Data Mining, Chicago, Illinois, USA, 21–24 August 2005*. New York: ACM, 2005, pp. 32–40.

53. Ganon, Z., Keinan, A., Ruppin, E. Evolutionary Network Minimization: Adaptive Implicit Pruning of Successful Agents. In: W. Banzhaf, et al. *Advances in Artificial Life: 7th European Conference, ECAL 2003*. Dortmund, *Germany, 14–17 September 2003*. Springer-Verlag, 2003, pp. 319–327. (Lecture Notes in Artificial Intelligence, vol. 2801).

54. Glorot, X., Bordes, A., Bengio, Y. Deep Sparse Rectifier Neural Networks. In: *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics (AISTATS), 2011, Fort Lauderdale, FL, USA, 11–13 April 2011*. Vol 15 of JMLR: W&CP 15, pp. 315–323.

55. Golik, P., Doetsch, P., Ney, H. Cross-Entropy vs. Squared Error Training: a Theoretical and Experimental Comparison. In: *Proc. of 14th Annual Conf. of the Intern. Speech Communication Association (Interspeech 2013), 25–29 August 2013, Lyon, France*. Lyon, 2013, pp. 1756–1760.

56. Goodfellow, I., Bengio, Y., Courville, A. *Deep Learning*, MIT Press, 2016, 785 p. Available from: http://www.deeplearningbook.org.

57. Goodfellow, I., Pouget-Abadie, J., Mirza, M., et al. Generative Adversarial Nets. In: *Advances in Neural Information Processing Systems*. Curran Associates, Inc. 2014, vol. 27, pp. 2672–2680. Available from: http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf.

58. Graves, A., Wayne, G., Reynolds, M., et al. Hybrid computing using a neural network with dynamic external memory. *Nature*. 2016, vol. 538, pp. 471–476.

59. Grossberg, S. Adaptive Resonance Theory: How a brain learns to consciously attend, learn, and recognize a changing world. *Neural Networks*. 2012, Special Issue. Available from: https://doi.org/10.1016/j.neunet.2012.09.017.

60. Guyon, I., Gunn, S., Nikravesh, M., Zadeh, L., eds. *Feature Extraction, Foundations and Applications*. Physica-Verlag, Springer, 2006. 781 p. (Series Studies in Fuzziness and Soft Computing).

61. Grünbaum, B. *Convex Polytopes*. 2nd ed. Springer–Verlag, 2003. 471 p. (Series: Graduate Texts in Mathematics, vol. 221)

62. Hahnloser, R. H. R., Sarpeshakar, R., Mahowald, M. A., et al. Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit. *Nature*. 2000, vol. 405, pp. 947–951.

63. Hailesilassie, T. Rule Extraction Algorithm for Deep Neural Networks: A Review. *International Journal of Computer Science and Information Security.* 2016, vol. 14, no. 7, pp. 376–381.

64. Hammer, B., Rechtien, A., Strickert, M., et al. Rule Extraction from Self-Organizing Networks. In: *Artificial Neural Networks — ICANN 2002: Proceedings of International*

Conference, 28–30 August 2002, Madrid, Spain. Berlin, Heidelberg: Springer, 2002, pp. 877–883. (Lecture Notes in Computer Science. Vol. 2415/2002).

65. Hammer, B., Villmann, T. Estimating Relevant Input dimensions for Self-organizing Algorithms. In: N. Allison, H. Yin, L. Allison, J. Slack, eds. *Advances in Self-Organizing Maps*. London: Springer-Verlag, 2001, pp. 173–180.
66. Han, S., Mao, H., Dally, W. J. *Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding* [online]. ArXiv.org, 2016 [viewed 1 June 2019]. Available from: https://arxiv.org/abs/1510.00149.
67. Hawkins, J., Ahmad, S., Cui, Y. A Theory of How Columns in the Neocortex Enable Learning the Structure of the World. *Frontiers in Neural Circuits*. 2017, vol. 11 (81). Available from: https://doi.org/10.3389/fncir.2017.00081 [Accessed 1 June 2019].
68. Hayashi, Y., Setiono, R., Yoshida, K. A Comparison between Two Neural Network Rule Extraction Techniques for the Diagnosis of Hepotibiliary Disorders. *Artificial Intelligence in Medicine*. 2000, vol. 20, No. 3, pp. 205–216.
69. He, K., Zhang, X., Ren, S., et al. *Deep Residual Learning for Image Recognition* [online]. arXiv.org, 2015 [viewed 1 September 2017]. Available from: https://arxiv.org/abs/1512.03385.
70. Hochreiter, S., Schmidhuber, J. Long Short-Term Memory. *Neural Computation*. 1997, vol. 9, No. 8, pp. 1735–1780.
71. Hruschka, E. R., Ebecken, N. F. F. Extracting Rules from Multilayer Perceptrons in Classification Problems: A Clustering-based Approach. *Neurocomputing*. 2006, vol. 70, no. 1-3, pp. 384–397.
72. Hu, Y. H., Hwang, J-N. Ch.1. Introduction to Neural Networks for Signal Processing. In: Y. H. Hu, J-N. Hwang, eds. *Handbook of Neural Network Signal Processing*. CRC Press, 2002, pp. 12–41.
73. Huysmans, J., Baesens, B., Vanthienen, J. *Using Rule Extraction to Improve the Comprehensibility of Predictive Models*. Leuven, Belgium: Katholieke Universiteit Leuven, 2006. 55 p.
74. Hyontai, S. More Balanced Decision Tree Generation for Imbalanced Data Sets including the Parkinson's Disease Data. *Intern. J. of Biology and Biomedical Engineering*. 2016, vol. 10, pp. 115–123.
75. Iqbal, R. A. Eclectic Rule Extraction from Neural Networks Using Aggregated Decision Trees. In: *7^th^ Intern. Conf. on Electrical & Computer Engineering (ICECE 2012), Dhaka, Bangladesh, 20–22 December 2012*. IEEE, 2013, pp. 1–5.
76. Jacobsson, H. Rule Extraction from Recurrent Neural Networks: A Taxonomy and Review. *Neural Computation*. 2005, vol. 17, no. 6, pp. 1223–1263.
77. Jifeng, D., Yang, L., Nian, W.Y., Generative Modeling of Convolutional Neural Networks. In: *3rd Intern. Conf. on Learning Representations* (*ICLR), New Orleans, USA, 6–9 May* 2015. Available from: https://arxiv.org/abs/1412.6296#_[Accessed 1 June 2019].
78. Ji, C., Snapp, R. R., Psaltis, D. Generalizing Smoothness Constraints from Discreet Samples. *Neural Computation*. 1990, vol. 2, no. 2, pp. 188–197.

79. Jumutcs, V., Bondarenko, A. Polytope Classifier: A Symbolic Knowledge Extraction from Piecewise-Linear Support Vector Machine. In: *Knowledge-Based and Intelligent Information and Engineering Systems: 15th Intern. Conf. (KES 2011): Proc., Part 1, Germany, Kaiserslautern, 12–14 September 2011*. 2011, pp. 62–71.

80. Nunez, H., Angulo, C., Catala, A. Rule Extraction from Support Vector Machines. In: *Proceedings of the 10th European Symposium on Artificial Neural Networks, Bruges, Belgium, 24–26 April 2002*. d-side publi., 2002. pp. 107–112. ISBN 2-930307-02-1.

81. Qiang, X., Cheng, G., Wang, Z. An Overview of Some Classical Growing Neural Networks and New Developments. In: *Proceedings of the 2nd International Conference on Education Technology and Computer (ICETC 2010), 22–24 June 2010, Shanghai, China*. Vol. 3. Chengdu, China: IEEE, 2010, pp. V3-351 – V3-355.

82. Quinlan, R. *C4.5: Programs for machine learning*. San Manteo, CA: Morgan Kaufman, 1993. 302 p.

83. Kahramanli, H., Allahverdi, N. Rule extraction from trained adaptive neural networks using artificial immune systems. *Expert Systems with Applications: An Intern. J. archive*. 2009, vol. 36, no. 2, pp. 1513–1522.

84. Kamruzzaman, S. M., Islam, M. M. An Algorithm to Extract Rules from Artificial Neural Networks for Medical Diagnosis Problems. *Intern. J. of Information Technology*. 2006, vol. 12, no. 8, pp. 41–59.

85. Karpathy, A., Fei-Fei, L. Deep Visual-Semantic Alignments for Generating Image Descriptions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2014, vol. 39, no. 4. Available from: arXiv:1412.2306, [viewed 1 June 2019].

86. Kasabov, N. Evolving Connectionist Systems: From Neuro-Fuzzy-, to Spiking- and Neuro-Genetic. In J. Kacprzyk, W. Pedrycz, eds. *Springer Handbook of Computational Intelligence.* Berlin Heidelberg: Springer, 2015, pp. 771–782. Available from: doi:10.1007/978-3-662-43505-2.

87. Kingma, D. P., Welling, M. *Auto-encoding Variational Bayes* [online]. arXiv.org, 2013 [viewed 1 June 2019]. Available from: https://arxiv.org/pdf/1312.6114.pdf.

88. Kohonen, T. Learning Vector Quantization. In: A.M. Arbib, ed. *The Handbook of Brain Theory and Neural Networks.* 2nd ed. Cambridge, London: The MIT Press, 2002, pp. 631–635.

89. Konohen, T. *Self-Organizing Maps*. 3rd ed. Berlin: Springer–Verlag, 2001. 528 p.

90. Krishnagopal, D. *The Biology of Thought: A Neuronal Mechanism in the Generetion oh Trought – A New Molecular Model*. Amsterdam: Academic Press, 2014. 248 p.

91. Krizhevsky, A. *Learning Multiple Layers of Features from Tiny Images*. Technical Report. University of Toronto, 2009. 60 p.

92. Krizhevsky, A., Sutskever, I., Hinton, G. Imagenet Classification with Deep Convolutional Neural Networks. In: *Advances in Neural Information Processing Systems (NIPS 2012)*. Vol. 25. MIT Press, 2012, pp. 1097–1105.

93. Henderson, E. K., Martinez, T. R. Constructing Low-Order Discriminant Neural Networks Using Statistical Feature Selection. *Journal of*

Intelligent Systems. 2007, vol. 16, no. 1, pp. 27–56. Available from https://doi.org/10.1515/JISYS.2007.16.1.27.

94. Langley, P., Laird, J. E., Rogers, S. Cognitive Architectures: Research Issues and Challenges. *Cognitive Systems Research*, 2009, vol. 10, no. 2, pp. 141–160. Available from: https://doi.org/10.1016/j.cogsys.2006.07.004.

95. LeCun, Y., Bottou L., Bengio, Y., et al. Gradient-based Learning Applied to Document Recognition. *Proceedings of the IEEE*. 1998, vol. 86, no. 11, pp. 2278–2324.

96. Liu, H., Setiono, R. Chi2: Feature Selection and Discretization of Numeric Attributes. In: *Proc. of the 7th IEEE Intern. Conf. on Tools with Artificial Intelligence, Herndon, Virginia 5-8 November 1995*. IEEE Computer Society Press, 1995, pp. 388–391.

97. Liu, H., Tan, S. T. X2R: A Fast Rule Generator. In: *Proc. of IEEE Intern. Conf. on Systems, Man and Cybernetics, Vancouver, BC, Canada, 22–25 October 1995*. IEEE Press, 1995, pp. 1631–1635.

98. Long, J., Shelhamer, E., Darrell, T. Fully Convolutional Networks for Semantic Segmentation. In: *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR 2015), Boston, MA, 7–12 June 2015*. IEEE, 2015, pp. 3431–3440.

99. Luger, G. F. *Artifical Intelligence: Structures and Strageies for Complex Problem Solving*. 6th ed. Pearson, 2008. 784 p.

100. Magdalena, L. Fuzzy Rule-Based Systems. In: J. Kacprzyk, W. Pedrycz, eds. *Springer Handbook of Computational Intelligence*. Berlin Heidelberg: Springer, 2015, pp. 203–218. Available from: DOI 10.1007/978-3-662-43505-2.

101. Manessi, F., Rozza, A., Bianco, S., Bianco, S., et al. Automated Pruning for Deep Neural Network Compression. In: *24th International Conference on Pattern Recognition (ICPR), Beijung, China, 20–24 August 2018*. IEEE, 2018. Available from: DOI:10.1109/ICPR.2018.8546129.

102. McCulloch, W., Pitts, W. A Logical Calculus of the Ideas Immanent in Nervous Activity. *Bulletin of Mathematical Biophysics*, 1943, vol. 5, pp. 115–133.

103. McGarry, K. J., Tait, J., Wermter, S., MacIntyre, J. Rule-Extraction from Radial Basis Function Networks. In: *International Conference on Artificial Neural Networks, Edinburgh, UK, 7–10 September 1999*. Vol. 2. London: Institution of Electrical Engineers, 1999, pp. 613–618.

104. Merino, E. R., Castrillejo, F. M., Pin, J. D., et al. *Weighted Contrastive Divergence* [online]. ArXiv.org, January 2018. Available from: https://arxiv.org/abs/1801.02567.

105. Minar, M. R., Naher, J. *Recent Advances in Deep Learning: An Overview* [online]. 2018 [viewed 1 June 2019]. Available from: https://arxiv.org/abs/1807.08169.

106. Mitchell, T**.** *Machine Learning.* McGraw Hill, 1997. 432 p.

107. Molchanov, P., Tyree, S., Karras, T., et al. Pruning Convolutional Neural Networks for Resource Efficient Inference. In: *Proceedings of 5th International Conference on Learning Representations (ICLR), Toulon, France, 24–26 April, 2017.* [Online]. Available from: https://arxiv.org/pdf/1611.06440.pdf.

108. Morissette, L., Chartier, S. The k-means Clustering Technique: General Considerations and Implementation in Mathematica. *Tutorials in Quantitative Methods for Psychology*. 2013, vol. 9, no. 1, pp. 15–24. Available from: DOI: 10.20982/tqmp.09.1.p015.

109. Mues, C., Baesens, B., Setiono, R., Vanthienen, J. From Knowledge Discovery to Implementation: A Business Intelligence Approach Using Neural Network Rule Extraction and Decision Tables. In: *Biennial Conference on Professional Knowledge Management*, 2005, pp. 483–495. Available from: DOI: 10.1007/11590019_55.

110. Nowlan, S. J., Hinton, G. E. Simplifying Neural Networks by Soft Weight-Sharing. *Neural Computation*. 1992, vol. 4, no. 4, pp. 473–493.

111. Omlin, C. W., Giles, C. L. Extraction of Rules from Discrete-time Recurrent Neural Networks. *Neural Networks*. 1996, vol. 9, no. 1, pp. 41–52.

112. Pengfei, Z., Qinghua, H. Rule extraction from support vector machines based on consistent region covering reduction. *Knowledge-Based Systems*. 2013, vol. 42, pp. 1–8. Available from: https://doi.org/10.1016/j.knosys.2012.12.003.

113. Phil, S. *Too Big to Ignore: The Business Case for Big Data*. Wiley, 2013. 231 p.

114. Prechelt, L. *Adaptive Parameter Pruning in Neural Networks*. International Computer Science Institute, Technical Report TR-95-009. Berkley, California: 1995. 14 p.

115. *Pytorch – Tensors and Dynamic Neural Networks in Python with Strong GPU Acceleration* [online]. PyTorch Team, 2017 [viewed 1 November 2017]. Available from: http://pytorch.org.

116. Ranzato, M. A., Poultney, C., Chopra, S., et al. Efficient Learning of Sparse Representations with an Energy-Based Model. In: B. Schölkopf, J. Platt, T. Hoffman, eds. *Advances in Neural Information Processing Systems 19*. Cambridge, MA : MIT Press, 2006, pp. 1137–1144.

117. Reed, R., Marks, R. J. *Neural Smithing: Supervised Learning in Feedforward Artificial Neural Networks*. MIT Press, 1999. 346 p.

118. Regulation (EU) 2016/679 of the European Parliament and of the Council [online]. *Official Journal of European Union*, 2016 [viewed 20 December 2017] Available from: http://ec.europa.eu/justice/data-protection/reform/files/regulation_oj_en.pdf.

119. Rumelhart, D.E., Hinton, G., Williams, R.J. Learning Representations by Back-propagating Errors. *Nature*. 1986, vol. 323, pp. 533–536.

120. Russakovsky, O., Deng, J., Su, H., et al. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*. 2015, vol. 115, no. 3, pp. 211–252.

121. Sato, A., Yamada, K. Generalized Learning Vector Quantization. In: D. S. Touretzky, M. C. Mozer, M. E. Hasselmo, eds. *Advances in Neural Information Processing Systems*. Vol. 8. Cambridge, MA, USA: MIT Press, 1996, pp. 423–429.

122. Sato M., Tsukimoto, H. Rule Extraction from Neural Networks via Decision Tree Induction. In: *Proc. IJCNN'01. Intern. Joint Conf. on Neural Networks, Washington, DC, 15–19 July 2001*. Vol. 3. IEEE, 2001, pp. 1870–1875.

123. *Scikit Learn Team, Choosing the Right Estimator* [online]. Scikit [viewed 19 May 2017]. Available from: http://scikit-learn.org/stable/tutorial/machine_learning_map/index.html.
124. Setiono, R., Baesens, B., Mues, C. Recursive Neural Network Rule Extraction for Data With Mixed Attributes. *IEEE Trans. on Neural Networks*. 2008, vol. 19, no. 2, pp. 299-307.
125. Setiono, R., Leow, W. H. Pruned Neural Networks for Regression. In: *Proceedings of PRICAI 2000, The 6th Pacific Rim International Conference on Artificial Intelligence,* Melbourne, Australia, 28 August – 1 September, 2000. Berlin: Springer-Verlag, 2000, pp. 500–509. (Lecture Notes in *Artificial Intelligence*. Vol. 1886). Available from: doi:10.1007/3-540-44533-1_51.
126. Setiono, R., Leow, W. K., Zurada, J. M. Extraction of Rules from Artificial Neural Networks for Nonlinear Regression. *IEEE Trans. on Neural Networks*. 2002, vol. 13, no. 3, pp. 564–577.
127. Setiono, R., Thong, J. An Approach to Generate Rules from Neural Networks for Regression Problems. *European J. of Operational Research.* 2004, vol. 155, no. 1, pp. 239–250.
128. Sethi, K. K., Mishra, D. K., Mishra, B. KDRuleEx: A Novel Approach for Enhancing User Comprehensibility Using Rule Extraction. In: *3rd International Conference on Intelligent Systems Modelling and Simulation (ISMS), Kota Kinabalu, Malaysia, 8–10 February 2012*. IEEE, 2012, pp. 55–60.
129. Sharma, A., Wolfe, N., Raj, B. *The Incredible Shrinking Neural Network: New Perspectives on Learning Representations through The Lens of Pruning* [online]. 2017 [viewed 1 June 2019]. Available from: https://arxiv.org/abs/1701.04465#
130. Shinde, P. Data Mining Using Artificial Neural Network Tree. *IOSR J. of Engineering,* 2012, pp. 9–12.
131. Siebel, N. T., Botel, J., Sommer, G. Efficient neural network pruning during neuro-evolution. In: *2009 International Joint Conference on Neural Networks Atlanta, GA, USA, 14–19 June 2009*. IEEE, 2009, pp. 2920–2927. Available from: doi:10.1109/IJCNN.2009.5179035.
132. Silver, D., Huang, A., Maddison, C. J., et al. Mastering the Game of Go with Deep Neural Networks and Tree Search. *Nature.* 2016, vol. 529, pp. 484–489.
133. Simonyan, K., Vedaldi, A., Zisserman, A. Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. In: *International Conference on Learning Representations, (ICLR 2014), Banff, Canada, 14–16 April 2014, Workshop Proceedings*. Available from: https://arxiv.org/abs/1312.6034.
134. Srivastava, N., Hinton, G., Krizhevsky, A., et al. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *J. of Machine Learning Research*, 2014, vol. 15, pp. 1929–1958.
135. Taur, J., Kung, S-Y., Lin, S-H. Hierarchical Fuzzy Neural Networks for Pattern Classification. In: Y. H. Hu, J-N. Hwang, eds. *Handbook of Neural Network Signal Processing.* CRC Press, 2002, pp. 348–381.

136. Theano Development Team, *Theano: A Python Framework for Fast Computation of Mathematical Expressions* [online]. arXiv e-prints, 2016 [viewed 1 November 2017]. Available from: http://arxiv.org/abs/1605.02688.

137. Thompson, M.–E. *NDCC: Normally Distributed Clustered Datasets on Cubes* [online]. Madison: University of Wisconsin, 2006 [viewed 15 September 2017]. Available from: http://www.cs.wisc.edu/dmi/svm/ndcc/.

138. Thrun, S. *The MONKs problems: A Performance Comparison of Different Learning Algorithms*. Technical Report CS-91-197. Pittsburg, PA: 1991.

139. Tickle A. B., Andrews R., Golea M., et al. The Truth will come to Light: Directions and Challenges in Extracting the Knowledge Embedded within Trained Artificial Neural Networks. *IEEE Trans. on Neural Networks*. 1998, vol. 9, no. 6, pp. 1057–1067.

140. Tickle, A. B., Orlowski, M. J, Diederich, J. DEDEC: A Methodology for Extracting Rules from Trained Artificial Neural Networks. In: *Proceedings of the Rule Extraction from Trained Artificial Neural Networks (AISB'96), Brighton, UK, 2 April 1996*. Brighton, University of Sussex, 1996, pp. 90–102.

141. Tsukimoto, H. Extracting Rules from Trained Neural Networks. *IEEE Transactions on Neural Networks*. 2000, vol. 11, no. 2, pp. 377–389.

142. Van der Maaten, L. J. P., Hinton, G. E. Visualizing High-Dimensional Data Using t-SNE. *J. of Machine Learning Research*. 2008, vol. 9, pp. 2579–2605.

143. Wan, W., Mabu, S., Shimada, K., Hirasawa, K. Enhancing the Generalization Ability of Neural Networks through Controlling the Hidden Layers. *Applied Soft Computing J.* 2009, vol. 9, no. 1, pp. 404–414.

144. Weigend, A. S., Rumelhart, D. E., Huberman, B. A. Generalization by Weight-Elimination Applied to Currency Exchange Rate Prediction. In: *Proceedings of IJCNN-91-Seattle International Joint Conference on Neural Networks, Seattle, Canada, 8–12 July 1991*. Vol. 2. IEEE, 1991, pp. 837–841.

145. Xing, H.-J., Hu, B.-G. Two Phase Construction of Multilayer Perceptrons using Information Theory. *IEEE Trans. on Neural Networks*. 2009, vol. 20, no. 4, pp. 715–721.

146. Xu, S., Chen, L. A Novel Approach for Determining the Optimal Number of Hidden Layer Neurons for FNN's and Its Application in Data Mining. In: D. Tien, M. Kavakli, eds. *5th Intern. Conf. on Information Technology and Applications (ICITA 2008), Cairns, Queensland, Australia, 23–26 June 2008*. [electronic resource]. Bathurst, N.S.W.: Macquarie Scientific Publishing, 2008, pp. 683–686. Available from: https://eprints.utas.edu.au/6995/1/02-au-xu.pdf.

147. Zeiler, M. D., Fergus, R. Visualizing and Understanding Convolutional Networks. In: D. Fleet, T. Pajdla, B. Schiele, T. Tuytelaars, eds. *Computer Vision - ECCV 2014: 13th European Conf., Zurich, Switzerland, 6–12 September 2014.* Part I. Springer, 2014, pp. 818–833. Available from: https://doi.org/10.1007/978-3-319-10590-1_53.

148. Zhang, H., Xu, T., Li, H., et al. *StackGAN: Text to Photo-realistic Image Synthesis with Stacked Generative Adversarial Networks* [online]. arXiv.org, 2016 [viewed 7 June 2017]. Available from: https://arxiv.org/pdf/1612.03242.pdf.

149. Zhang, Z., Qiao, J. A Node Pruning Algorithm for Feedforward Neural Network Based on Neural Complexity. In: *Proceedings of 2010 Internation al Conference on Intelligent Control and Information Processing (ICICIP), Dalian, China, 13–15 August 2010*. IEEE, 2010, pp. 406–410.

150. Zilke, J. R., Mencia, E. L., Janssen, F. DeepRED–Rule Extraction from Deep Neural Networks. In: T. Calders, M. Ceci, D.Malerba, eds. *Discovery Science: Proceedings of the 19th International Conference, DS 2016, Bari, Italy, 19–21 October 2016*. Springer, 2016, pp. 457–473 (Lecture Notes in Artificial Intelligence. Vol. 9956). Available from: http://dx.doi.org/10.1007/978-3-319-46307-0_29.