

IMPROVEMENT OF DEEP REINFORCEMENT
MODELS USING EXTREME LEARNING MACHINE
FOR AUTONOMOUS AGENTS IN UNSTRUCTURED
ENVIRONMENT

BY

NOUAR ALDAHOUL

A thesis submitted in fulfilment of the requirement for the
degree of Doctor of Philosophy (Engineering)

Kulliyyah of Engineering
International Islamic University Malaysia

MARCH 2021

ABSTRACT

Creating an autonomous agent, that gets real observations such as sensory data and images from the surrounding environment and learns optimal sequential actions, has been considered as one of the main goals of Artificial General Intelligence (AGI). Deep (Hierarchical) Reinforcement Learning (HRL/DRL) can address this objective. Traditional deep reinforcement learning methods suffer from long learning and training time resulted from the need to fine-tune the weights iteratively in the network. This research investigates the previous problem by utilizing a random weights generation approach that is based on Extreme Learning Machine. This method benefits from the randomness of input weights and least square solution in output weights calculation to reduce the training time by an order of magnitude. Hierarchical ELM (H-ELM) and Local Receptive Field ELM (LRF-ELM) are recent versions of multilayer ELM to respectively learn and extract features by hierarchical learning scheme. They have outperformed other existing deep models in terms of learning time (speed). H-ELM's architecture was found to be similar to gradient-based (GB) auto-encoder without weights fine-tuning. However, H-ELM gives higher learning speed compared to the GB autoencoder. Moreover, LRF-ELM was found as similar to Convolutional Neural Network (CNN) without weights fine-tuning. It has outperformed the traditional CNN in the term of learning time. Therefore, in this research, the proposed method, which combines RL with H-ELM or LRF-ELM, is an efficient solution to approximate the action-value function and learn an optimal policy directly from visual data (images) in a short time. In addition, this research proposed a novel method called Convolutional H-ELM (CH-ELM) which is a combination of pre-trained CNN and H-ELM. This method has outperformed either CNN or H-ELM in terms of accuracy and RMSE. The experimental results have been analyzed and evaluated in different applications such as target reaching arm, 2D maze navigation, slide puzzle game, objects sorting, and rock-paper-scissor game. The data samples have been trained and tested to investigate the robustness of the proposed systems. It was found that the proposed models can reduce the learning time by an order of magnitude in various tasks without degrading the performance. The big improvement in learning speed in the proposed method can neglect the slight drop in accuracy in few tasks compared to traditional methods. Therefore, the proposed method can balance the trade-off between learning speed and good performance. In addition, it is able to run on traditional CPUs that are available in the most of the low cost embedding systems.

خلاصة البحث

تعد عملية إيجاد عميل ذاتي القرار قادر على رصد المشاهدات الحقيقية كبيانات الحساسات و الصور من البيئة المحيطة وتعلم سلسلة من الأفعال المثالية من أهم أهداف الذكاء العام الصناعي. استطاع التعلم المعزز الهرمي (العميق) تحقيق هذا الهدف. تعاني الطرق التقليدية للتعلم المعزز العميق من طول زمن التعلم والتدريب الناتج من الحاجة إلى توليف الأوزان بشكل متكرر في الشبكة. في هذا البحث تم دراسة هذه المشكلة بالاستفادة من مفهوم توليد الأوزان العشوائية القائم على خوارزمية ELM. هذه الطريقة تستفيد من عشوائية أوزان الدخل ومن الحل القائم على المربعات الصغرى في حساب أوزان الخرج لانقاص زمن التدريب عدد من المرات. تم الاستفادة من البنية الهرمية H-ELM و حقول الاستقبال المحلي ELM-LRFs وهما إصداران حديثان لشبكة ELM متعددة الطبقات ويتم فيهما تعلم الميزات أو استخراجها عن طريق التعلم الهرمي. هذه النماذج تفوقت على نماذج التعلم العميق الموجودة مسبقاً من خلال زمن التعلم (سرعة التعلم). إن بنية H-ELM تشبه المرمز الألي القائم على هبوط الانحدار (Gradient Descent based auto encoder) ولكن بدون الحاجة إلى التوليف. ومع ذلك فإن البنية H-ELM تتمتع بسرعة تدريب أفضل مقارنة مع الأخير. كما تم استخدام حقول الاستقبال المحلي كبنية بديلة لمشابهة للشبكات العصبونية الالتفافية CNN ولكن بدون الحاجة إلى توليف الأوزان وضبطها. وقد تم إثبات تفوقها على CNN من حيث سرعة التعلم. لذا فإن الطريقة المقترحة في هذا البحث والتي تعتمد على دمج التعلم المعزز مع H-ELM أو ELM-LRFs هي حل فعال لتقريب تابع قيم الأفعال (Action Value Function) و تعلم الاستراتيجية المثالية بشكل مباشر من المعطيات المرئية (الصور) كمدخل للنظام خلال زمن قصير. بالإضافة لما سبق تم في هذا البحث اقتراح طريقة جديدة تدعى CH-ELM و الذي تم فيه دمج الشبكة الالتفافية المدربة مسبقاً مع الشبكة الهرمية العشوائية H-ELM و قد اثبت هذا النموذج تفوقه على كل من الشبكة الالتفافية CNN و شبكة H-ELM من حيث الدقة وجذر متوسط مربع الخطأ. تم في هذا البحث تحليل وتقييم النتائج التجريبية في تطبيقات مختلفة كتطبيق ذراع روبوتية يبحث عن الهدف وعميل في متاهة ثنائية البعد ولعبة البازل المتزحلِق وفرز الأغراض المختلفة ولعبة حجر ورق مقص. تم تدريب وفحص عينات من البيانات للتأكد من متانة النظام المقترح. وجد أن النموذج المقترح قادر على انقاص زمن التعلم عدد من المرات في مهام مختلفة دون تراجع مستوى الأداء. إن التحسن الكبير في سرعة التعلم في الطريقة المقترحة سمح بإهمال التراجع الطفيف في الدقة في بعض المهام بالمقارنة مع الطرق التقليدية. لذلك فإن الطريقة المقترحة تستطيع موازنة المقايضة بين سرعة التعلم والاداء الجيد. بالإضافة إلى أنها قابلة للتنفيذ على المعالجات التقليدية المتوفرة في معظم الأنظمة المضمنة منخفضة التكلفة.

APPROVAL PAGE

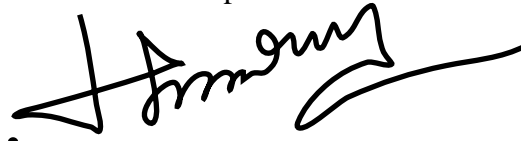
The thesis of Nour AlDahoul has been approved by the following:



on behalf Zaw Zaw Htike
Supervisor



Amir Akramin bin Shafie
Co-Supervisor



Md. Raisuddin Khan
Internal Examiner

Sazali Bin Yaacob
External Examiner

Mohd Rizon Mohamed Juhari
External Examiner

Imad Fakhri Taha Alshaikhli
Chairman

DECLARATION

I hereby declare that this thesis is the result of my own investigations, except where otherwise stated. I also declare that it has not been previously or concurrently submitted as a whole for any other degrees at IIUM or other institutions.

Nouar AIDahoul

SignatureNouar AIDahoul.....

Date 10/02/2021

INTERNATIONAL ISLAMIC UNIVERSITY MALAYSIA

**DECLARATION OF COPYRIGHT AND AFFIRMATION OF
FAIR USE OF UNPUBLISHED RESEARCH**

**IMPROVEMENT OF DEEP REINFORCEMENT MODELS
USING EXTREME LEARNING MACHINE FOR
AUTONOMOUS AGENTS IN UNSTRUCTURED
ENVIRONMENT**

I declare that the copyright holders of this thesis are jointly owned by the student and IIUM.

Copyright © 2021 Nouar AlDahoul and International Islamic University Malaysia. All rights reserved.

No part of this unpublished research may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise without prior written permission of the copyright holder except as provided below

1. Any material contained in or derived from this unpublished research may be used by others in their writing with due acknowledgement.
2. IIUM or its library will have the right to make and transmit copies (print or electronic) for institutional and academic purposes.
3. The IIUM library will have the right to make, store in a retrieved system and supply copies of this unpublished research if requested by other universities and research libraries.

By signing this form, I acknowledged that I have read and understood the IIUM Intellectual Property Right and Commercialization policy.

Affirmed by Nouar AlDahoul

.....Nouar AlDahoul.....
Signature

10/02/2021
Date

ACKNOWLEDGEMENTS

My first word of thanks goes to ALLAH Who gave me the chance and the health to accomplish this work.

I do not forget to thank Assoc professor Zaw Zaw Htike who guided me throughout the path of this research. Additionally, I also thank Prof. Md. Raisuddin Khan and Prof. Amir Akramin Shafie who supervised me after VIVA and helped to improve the quality of this work.

Finally, this study should not have seen the light without the efforts and support given to me by my mother. I give many thanks to her and dedicate this modest work to her. She was so patient with me in this journey and encouraged me to accomplish this work.

TABLE OF CONTENTS

Abstract	ii
Abstract in Arabic	iii
Approval Page.....	iv
Declaration	v
Copyright Page.....	vi
Acknowledgements	vii
Table of Contents	viii
List of Tables	xii
List of Figures	xv
List of Symbols	xx
List of Abbreviations	xxi
CHAPTER ONE: INTRODUCTION	1
1.1 Research Background	1
1.2 Problem Statement.....	4
1.3 Research Questions.....	5
1.4 Research Objectives.....	5
1.5 Research Hypotheses	6
1.6 Research Methodology	6
1.7 Research Philosophy.....	9
1.8 Significance of the Study.....	10
1.9 Research Scope	10
1.10 Contributions	11
1.11 Thesis Organization	16
CHAPTER TWO: LITERATURE REVIEW	18
2.1 Introduction.....	18
2.2 Unstructured Environment.....	21
2.3 Classical Reinforcement Learning.....	24
2.4 Hidden State.....	28
2.5 Continous States in RL	30
2.6 Dimensionality And Space Explosion In RL	31
2.7 Model-Based vs Model-Free Reinforcement Learning	32
2.8 Exploration and Exploitation Dilemma	34
2.9 Value Iteration vs Policy Iteration	35
2.10 Partially Observable Markov Decision Process	36
2.11 Dynamic Programming (DP).....	39
2.12 Monte Carlo (MC) Method.....	44
2.13 Temporal Difference (TD).....	45
2.13.1 Q Learning and Sarsa	46
2.13.1.1 Q Learning	46
2.13.1.2 Sarsa.....	47
2.13.2 Actor-Critic Methods.....	47
2.14 Function Approximation of Value Function	49
2.15 Batch Reinforcement Learning.....	52

2.16 Fitted Value Iteration Approach and Applications.....	55
2.17 Shallow Extreme Learning Machine	57
2.17.1 Batch ELM for Classification and Regression	58
2.17.2 Online Sequential ELM (OSELM).....	61
2.17.3 Kernel ELM for Classification	63
2.18 ELM based Reinforcement Learning.....	63
2.19 Deep Learning	65
2.19.1 Deep Neural Network.....	66
2.19.2 Convolutional Neural Network.....	67
2.19.3 Pre-trained Convolutional Neural Network.....	73
2.19.4 Gradient-based Autoencoder	75
2.19.5 Stacked Autoencoder	76
2.20 Deep Extreme Learning Machine	77
2.20.1 Hierarchical Extreme Learning Machine.....	80
2.20.1.1 H-ELM for Feature Learning.....	80
2.20.1.2 Traditional ELM based Autoencoder	82
2.20.1.3 Sparse ELM based Autoencoder	83
2.20.2 Local Receptive Field-based ELM	86
2.20.2.1 Hidden Neurons in Full and Local Connections.....	86
2.20.2.2 LRF based ELM Implementation	89
2.20.2.3 LRF Capability for Approximation and Classification ...	91
2.21 Deep Reinforcement Learning.....	93
2.21.1 Deep RL in Continuous Space.....	99
2.22 Learning Time in Deep Reinforcement Learning.....	100
2.23 Summary.....	102

CHAPTER THREE: METHODS AND ALGORITHMS.....104

3.1 Introduction.....	104
3.2 The Proposed Methods And Models	105
3.2.1 Batch Extreme Learning Machine as Function Approximation ...	105
3.2.2 Online Sequential Extreme Learning Machine as Function Approximation.....	106
3.2.3 Kernel Extreme Learning Machine as Function Approximation...	106
3.2.4 Deep Extreme Learning Machine as Feature Learner	106
3.2.5 Q Learning as Optimal Control Algorithm.....	107
3.2.6 Experience Replay in Reinforcement Learning.....	109
3.2.7 ELM Fitted Q Iteration	109
3.2.8 Deep Reinforcement Learning Control Algorithm.....	110
3.2.9 Experience Replay in Deep Reinforcement Learning	111
3.3 The Proposed Agents.....	112
3.3.1 Extreme Learning Machine based RL with Experience Replay...	114
3.3.2 The Proposed H-ELM based Q Learning Agent	120
3.3.3 The Proposed SH-ELM based Q Learning Agent	123
3.3.4 The Proposed LRF-ELM based Q Learning Agent.....	127
3.3.5 The Proposed Convolutional H-ELM Learning Model.....	129
3.3.6 The Proposed Convolutional H-ELM based Q Learning Agent ...	130
3.3.7 Learning All Actions with Single Forward Pass	131
3.3.8 Novel Q-function Formulation for Specific Tasks	132
3.3.9 The RL-HELM based Ensemble Model for Selection of	

Feature Learners	133
3.3.10 The RL based Ensemble Model for Selection of Regressors	135
3.3.11 Various Feature Engineering Models	136
3.3.11.1 Static Image in Low Dimensional Space with One Action	137
3.3.11.2 Static Image with All Actions	137
3.3.11.3 Dynamic Visual Data with One Action	138
3.3.11.4 Dynamic Visual Data with All Actions	139
3.4 Summary	140
CHAPTER FOUR: EXPERIMENTAL DESIGN AND RESULTS	142
4.1 Introduction.....	142
4.2 Unstructured environment and data.....	145
4.3 Human Activity Recognition (HAR) Application.....	146
4.3.1 Data Presentation	147
4.3.2 Dataset	148
4.3.3 Activity Images.....	149
4.3.4 The Proposed Architecture of H-ELM	149
4.3.5 Accuracy Analysis	150
4.3.6 Speed Analysis.....	153
4.3.7 Feature Fusion with Q-Learning.....	154
4.4 Regression Applications	155
4.4.1 Image Rotation Angle Prediction	157
4.4.2 Arm's End Effector.....	157
4.4.3 Architectures and Hyperparameters	159
4.4.3.1 CNN Architecture.....	159
4.4.3.2 Pre-trained CNN Model.....	162
4.4.3.3 LRF-ELM Architecture	163
4.4.3.4 H-ELM Architecture.....	164
4.4.4 Results Analysis and Comparison	164
4.4.4.1 Accuracy Analysis	165
4.4.4.2 Speed Analysis	169
4.5 The Proposed Learning Agents	171
4.5.1 H-ELM Based Q Learning for Maze Navigation Application	171
4.5.1.1 Accuracy Analysis	175
4.5.1.2 Speed Analysis	178
4.5.1.3 Policy Transfer	179
4.5.2 H-ELM Based Q Learning for Object Sorting	181
4.5.2.1 Gripper Angle Selection	182
4.5.2.2 Shape Sorting.....	183
4.5.2.2.1 Reward Function as A Binary Classifier	185
4.5.2.3 Size Sorting.....	187
4.5.2.4 Colour Sorting	188
4.5.2.5 Training Environment.....	189
4.5.2.6 Dataset	190
4.5.2.7 Feature Learning and Hyperparameters.....	191
4.5.2.8 Experimental Protocol and Results.....	192
4.5.3 Rock-Paper-Scissor Game	199
4.5.3.1 H-ELM-Q learning Agent	200

4.5.3.1.1 Accuracy Analysis	202
4.5.3.1.2 P-value for The First Hypothesis	204
4.5.3.1.3 Speed Analysis.....	206
4.5.4 LRF-ELM-Q Agent	208
4.5.5 Slide Puzzle Game	210
4.5.5.1 Results Analysis for Numeric 3-Puzzle.....	212
4.5.5.2 Results Analysis for Visual 5-Puzzle	217
4.5.6 Sequential H-ELM vs H-ELM	220
4.5.6.1 Sequential H-ELM-Q Agent vs H-ELM-Q Agent.....	221
4.5.7 RL based Ensemble Model for Feature Selection	222
4.5.8 Convolutional Hierarchical Extreme Learning Machine (CH-ELM)	227
4.5.8.1 LHI-Animal Face Dataset	228
4.5.8.2 ETH-80 Dataset	228
4.5.8.3 Accuracy and Speed Analysis	230
4.5.8.4 P-value for The Second and Third Hypotheses	235
4.6 Research Hypotheses Discussion	237
4.7 Summary.....	239
CHAPTER FIVE: CONCLUSIONS AND RECOMMENDATIONS	242
5.1 Conclusions	242
5.2 Limitations.....	245
5.3 Recommendations	246
REFERENCES.....	248
APPENDIX A: The Code of The ELM Autoencoder Algorithm	263
APPENDIX B: The Derivation of Formulas and Mathematical Equations of The Sequential ELM Autoencoder	264
PUBLICATIONS	267

LIST OF TABLES

Table 2.1 Comparison Between Various Deep Reinforcement Learning in Terms of Training Time	102
Table 3.1 The Main Loop of ELM-Fitted Q	118
Table 3.2 The Main Loop of OSELM-Fitted Q	119
Table 3.3 Algorithm of H-ELM-Q Learning with Experience Replay	122
Table 3.4 Algorithm SH-ELM-Q Learning with Experience Replay	125
Table 3.5 Algorithm LRF-ELM-Q Learning with Experience Replay	128
Table 4.1 Comparison between State-of-The-Art Deep Models and H-ELM in Terms of Accuracy	152
Table 4.2 Comparison between H-ELM and SAEs in Terms of Time Efficiency	154
Table 4.3 The Architecture of S-CNN	160
Table 4.4 The Hyperparameters of S-CNN	162
Table 4.5 The Hyperparameters of AlexNet	163
Table 4.6 The Architecture and Hyperparameters of LRF-ELM	163
Table 4.7 The Architecture and Hyperparameters of H-ELM	164
Table 4.8 Comparison between Deep Models for Digit Rotation Regression Task	166
Table 4.9 Comparison Between SVM and H-ELM Added after AlexNet	167
Table 4.10 Comparison between Deep Model for End-Effector Regression Task	167
Table 4.11 Comparison between Deep Models in terms of Training Speed	170
Table 4.12 Accuracy of Testing Noisy Images with Different Model Architectures	177
Table 4.13 Comparison between PCA and H-ELM with Different Number of Features in Terms of Accuracy	178
Table 4.14 Comparison between H-ELM and Traditional Stacked Autoencoder in Terms of Training Time	179

Table 4.15 Comparison between Pure H-ELM-Q and H-ELM-Q with Policy Transfer in Terms of Training Time	181
Table 4.16 Average Accuracy of Size Sorting with 200 Training Samples	193
Table 4.17 Average Accuracy of Size Sorting with 500 Training Samples	193
Table 4.18 Average Accuracy of Shape Sorting with 500 Training Samples	194
Table 4.19 Average Accuracy of Orientation Sorting with 500 Training Samples	195
Table 4.20 Comparison between Pure H-ELM and H-ELM-Q learning in Terms of Accuracy	202
Table 4.21 Comparison between Different Architectures in Terms of Accuracy	204
Table 4.22 t-Test Paired Two Samples for Means (First Hypothesis)	205
Table 4.23 Comparison Between H-ELM and CNN in Terms of Accuracy	205
Table 4.24 Comparison Between H-ELM and CNN in Terms of Learning Time in One Episode	206
Table 4.25 Comparison Between H-ELM and CNN in Terms of Learning Time in 100 Episodes	206
Table 4.26 Comparison Between Different Architectures in Terms of Training Time	207
Table 4.27 The Architecture and Hyperparameters of LRF-ELM	209
Table 4.28 The Training Time of LRF-ELM with Different Number of Feature Maps	210
Table 4.29 The Policy Learning Time of LRF-ELM-Q learning with Different Number of Feature Maps	210
Table 4.30 The Architectures of CNN and H-ELM	215
Table 4.31 Comparison Between CNN-Q and H-ELM-Q in Terms of Training Time for Numeric 3-Puzzle	216
Table 4.32 Comparison Between CNN and H-ELM in Terms of Training Time for Visual 5-Puzzle	219
Table 4.33 Comparison Between CNN-Q and H-ELM-Q in Terms of Policy Learning Time for Visual 5-Puzzle	219
Table 4.34 Comparison Between H-ELM and SH-ELM in Terms of Training Time	220

Table 4.35 Comparison Between H-ELM and SH-ELM in Terms of Policy Learning Time	222
Table 4.36 Policy Learning Time of H-ELM with Different Numbers of Hidden Nodes	222
Table 4.37 The Comparison between E-H-ELM-Q and Ten Supervised H-ELM in Terms of RMSE (First Run)	224
Table 4.38 The Comparison between E-H-ELM-Q and Ten Supervised H-ELM in Terms of RMSE (Second Run)	225
Table 4.39 The Comparison between E-H-ELM-Q and Ten Supervised H-ELM in Terms of RMSE (Third Run)	226
Table 4.40 RMSE of H-ELM with Different Architectures and Number of Samples	227
Table 4.41 Training Time of H-ELM with Different Number of Samples	227
Table 4.42 Number of Training and Testing Samples in Each Dataset	230
Table 4.43 Comparison between S-CH-ELM and U-CH-ELM in Terms of Accuracy for LHI-Animal Face Dataset	231
Table 4.44 Comparison between S-CH-ELM and U-CH-ELM in Terms of Accuracy for ETH-80 Dataset	231
Table 4.45 Comparison Between Resnet50+SVM and CH-ELM in Terms of Accuracy for LHI-Animal Face Dataset	233
Table 4.46 Comparison Between Resnet50+SVM and CH-ELM in Terms of Accuracy for ETH-80 Dataset	233
Table 4.47 Comparison Between U-Resnet50 and U-CH-ELM in Terms of Accuracy for LHI-Animal Face Dataset	234
Table 4.48 Comparison Between U-Resnet50 and U-CH-ELM in Terms of Accuracy for ETH-80 Dataset	234
Table 4.49 Comparison Between U-Resnet50 and U-CH-ELM in Terms of Training Time for LHI-Animal Face Dataset	235
Table 4.50 Comparison Between U-Resnet50 and U-CH-ELM in Terms of Training Time for ETH-80 Dataset	235
Table 4.51 Comparison Between H-ELM and Gradient-based Agents in Terms of Learning Time	236
Table 4.52 t-Test Paired Two Samples for Means (Second Hypothesis)	236

LIST OF FIGURES

Figure 1.1 The Model of Interaction between Agent and Unstructured Environment	3
Figure 1.2 The Process Flow Showing The Research Methodology of This Work	8
Figure 1.3 The Process Flow Showing The Literature Review and Problem Definition Stages	9
Figure 1.4 The Block Diagram of The Proposed H-ELM-Q Learning Agent	12
Figure 1.5 The Block Diagram of The Proposed LRF-ELM-Q Learning Agent.	12
Figure 1.6 The Block Diagram of The CNN-H-ELM and CNN-SVM	13
Figure 1.7 The Block Diagram of The Convolutional Hierarchical ELM Q Learning Agent	14
Figure 1.8 The Block Diagram of The Sequential H-ELM-Q Learning Agent.	15
Figure 2.1 The Artificial General Intelligence Architecture	19
Figure 2.2 The Learning Agent in an Unstructured Environment	25
Figure 2.3 The Graphical Representation of HMM	30
Figure 2.4 The POMDP Diagram	38
Figure 2.5 The Difference between Reinforcement Learning and Planning	40
Figure 2.6 The Actor-Critic Architecture	49
Figure 2.7 The Graphical Sketch of The Batch RL Framework	53
Figure 2.8 The Graphical Sketch of The Deep Batch RL Framework	55
Figure 2.9 The Architecture of Deep Neural Network	67
Figure 2.10 The Structure of Supervised-CNN Model	72
Figure 2.11 The Structure of AlexNet-CNN Model	74
Figure 2.12 Single Autoencoder in The Training Stage	76
Figure 2.13 Deep Model based on A Stack of Gradient Autoencoders	76
Figure 2.14 The H-ELM learning algorithm including (A) Overall framework,	

(B) Implementation of ELM based Autoencoder. (C) Layout of One Single Layer inside The H-ELM.	85
Figure 2.15 ELM Hidden Node in Full Connections	87
Figure 2.16 ELM Hidden Node in Local Connections: Random Connections Generated due to Various Continuous Distributions	87
Figure 2.17 The Combinatorial Node of ELM : A Hidden Node Which Is A Subnetwork of Many Nodes With Linear or Nonlinear Pooling	88
Figure 2.18 The Architecture of LRF-ELM	92
Figure 3.1 The Learning Agent in an Unstructured Environment	108
Figure 3.2 The Block Diagram of The Reinforcement Learning Control System	111
Figure 3.3 The Detailed Block Diagram of The DRL Control System	114
Figure 3.4 The Block Diagram of CH-ELM	130
Figure 3.5 The Block Diagram of RL-HELM based Ensemble Model for feature learner selection	134
Figure 3.6 The Block Diagram of RL-HELM based Ensemble Model for Regressor Selection	136
Figure 3.7 The Block Diagram of Static Image in Low Dimensional Space with One Action	137
Figure 3.8 The Block Diagram of Static Image in Low Dimensional Space with Multiple Actions	138
Figure 3.9 The Block Diagram of Dynamic Data in Low Dimensional Space with One Action	139
Figure 3.10 The Block Diagram of Dynamic Data in Low Dimensional Space with Multiple Actions	140
Figure 4.1 Block Diagram of The Proposed System	148
Figure 4.2 Various Activity Images for Different Activities	149
Figure 4.3 Hierarchical Kernel ELM based Model for HAR System	150
Figure 4.4 The Confusing Matrix of The Testing Stage	152
Figure 4.5 The Comparison between State-of-The-Art and The Proposed Models in HAR	153
Figure 4.6 The comparison between H-ELM and SAEs in Terms of Training	

Speed	154
Figure 4.7 The Block Diagram used for Methods Comparison	156
Figure 4.8 Samples of Images in Rotated and Corrected Forms	157
Figure 4.9 Samples of Arm in Different Configurations	159
Figure 4.10 Feature Maps of Three Convolutional Layers: Conv1, Conv2, Conv4 for Digit Rotation with S-CNN	161
Figure 4.11 Feature Maps of Two Convolutional Layers: Conv1 and Conv4 for End -Effector Task with S-CNN	162
Figure 4.12 The Actual and Predicted Positions in 2D Space	165
Figure 4.13 A Residual Box Plot for Each Digit with CH-ELM Model	168
Figure 4.14 A Residual Box Plot for Each Position's Coordinate with CH-ELM Model	168
Figure 4.15 15 An Example of 4×4 Grid World	172
Figure 4.16 Image of The Maze without and with Noise	174
Figure 4.17 Training Images including 36 Images without Noise	174
Figure 4.18 Testing Images including 36 Noisy Images	175
Figure 4.19 Confusion Matrix for Testing The Maze Model	176
Figure 4.20 Average Predicted Action Value Curve	176
Figure 4.21 PCA Eigenvalues Curve	178
Figure 4.22 Various Environments in Maze Navigation Task	180
Figure 4.23 The Block Diagram of Object Location/ Orientation Sorting System	182
Figure 4.24 Sorting Objects According to Their Orientations	183
Figure 4.25 Sorting Objects According to Their Shapes	184
Figure 4.26 Samples of Difference Images; (A) Training Samples (B) Testing Samples (With Distortion)	186
Figure 4.27 Reward Binary Classifier Vs Multi-Classes Shape Classifier	186
Figure 4.28 Sorting Objects According to Their Sizes	187
Figure 4.29 Sorting Objects According to Their Color	188

Figure 4.30 The Working Field of The Sorting System	190
_Toc531520427Figure 4.31 Various Samples of The Objects for Sorting Task	191
Figure 4.32 H-ELM- Q-Learning Vs Supervised H-ELM for Shape Sorting	195
Figure 4.33 H-ELM- Q-Learning Vs Supervised H-ELM for Size Sorting	196
Figure 4.34 H-ELM- Q-Learning Vs Supervised H-ELM for Color Sorting	196
Figure 4.35 H-ELM- Q-Learning Vs Supervised H-ELM for Orientation Sorting	197
Figure 4.36 The Diagram of Sorting Object with Four Attributes Utilizing H-ELM-Q Learning Agent	199
Figure 4.37 Various Samples of The Rock-Paper-Scissor Dataset	201
Figure 4.38 Three Testing Samples from a Camera with Different Backgrounds	201
Figure 4.39 Comparison between Pure H-ELM and H-ELM-Q Agent	203
Figure 4.40 Comparison between Different Architectures in Terms of Training Time	207
Figure 4.41 Comparison between Pure LRF-ELM and LRF-ELM-Q Agent	209
Figure 4.42 Various Samples of 3-Puzzle	211
Figure 4.43 Ten 3×2 Image-Puzzles	212
Figure 4.44 Two Samples of Image-Puzzle in Random and Correct Orders	212
Figure 4.45 The Average Predicted Action-Value of H-ELM-Q Agent with Different Learning Rates	213
Figure 4.46 The Average Action-Value CNN-Q with Initial Learning Rate = 10-3 and batch size = 2	215
Figure 4.47 The Average Action-Value of CNN-Q with Initial Learning Rate = 10-5	216
Figure 4.48 The Average Action-Value of CNN-Q with Different Batch Sizes	216
Figure 4.49 The Average predicted Action-Value for 2 × 3 Image Puzzle with H-ELM-Q Agent	219
Figure 4.50 The Comparison between Output Weights of H-ELM and SH-ELM	221
Figure 4.51 The Comparison between E-H-ELM-Q and Ten Supervised H-ELM in Terms of RMSE (First Run)	224
Figure 4.52 The Comparison between E-H-ELM-Q and Ten Supervised	

H-ELM in Terms of RMSE (Second Run)	225
Figure 4.53 The Comparison between E-H-ELM-Q and Ten Supervised H-ELM in Terms of RMSE (Third Run)	226
Figure 4.54 Samples of LHI-Animal Face Dataset	229
Figure 4.55 Samples of ETH-80 Dataset	230

LIST OF SYMBOLS

a	Current Action
$E(\Theta)$	Loss Function
H^\dagger	Moore Penrose Inverse
H	Hidden Layer Output
Q	Action Value Function
R	Immediate Reward
s	Current State
s'	Next State
T	Target
V	State Value Function
w	Input Weights
α	Learning Rate
β	Output Weights
γ	Discount Factor
λ	Regularization Coefficient
π	Policy
ϵ	probability of greedy policy
Θ_k	Network's Parameters

LIST OF ABBREVIATIONS

A3C	Asynchronous Advantage Actor-Critic
AGI	Artificial General Intelligence
AI2THOR	An Interactive The House Of inteRactions
ANN	Artificial Neural Network
AUV	Autonomous underwater vehicle
BP	Back Propagation
BRL	Bayesian Reinforcement Learning
CH-ELM	Convolutional Hierarchical Extreme Learning Machine
CNN	Convolutional Neural Network
CPU	Central Processing Unit
DBM	Deep Boltzmann Machine
DBN	Deep Belief Network
DCNN	Deep Convolutional Neural Network
DDPG	Deep Deterministic Policy Gradient
DFQ	Deep Fitted Q
DNN	Deep Neural Network
DP	Dynamic Programming
DQN	Deep Q Network
DRL	Deep Reinforcement Learning
FA	Function Approximation
FASTA	Fast Iterative Shrinkage Thresholding Algorithm
GPU	Graphical Processing Unit
HAR	Human Activity Recognition
HMM	Hidden Markov Model
HRL	Hierarchical Reinforcement Learning
ICA	Independent Component Analysis
LRF-ELM	Local Receptive Field Extreme Learning Machine
LSPI	Least Square Policy Iteration
LSTD	Least Square Temporal Difference
LSTM	Long Short-Term Memory
MC	Monte Carlo
MDP	Markov Decision Process
ML	Machine Learning
MLELM	Multi-Layer Extreme Learning Machine
MLFN	Multi-Layer Forward Network
MLP	Multilayer Perceptron
NFQ	Neural Fitted Q Iteration

OS-ELM	Online Sequential Extreme Learning Machine
PCA	Principle Component Analysis
PDF	Probability Distribution Function
PGRL	Policy Gradient Reinforcement Learning
POMDP	Partial Observable Markov Decision Process
PPO	Proximal Policy optimization
RESNET	Residual Networks
RAE- ELM	Robust AdaBoost.RT based Ensemble Extreme Learning Machine
RIWG	Random Inputs Weight Generation
RL	Reinforcement Learning
RMSE	Root Mean Square Error
RVB LSPI	Least Squares Policy Iteration based on Random Vector Basis
RWG	Random Weight Generation
SAE	Stacked Autoencoder
SARSA	State Action Reward State Action
S-CNN	Supervised Convolutional Neural Network
SDA	Stacked Denoising Autoencoder
SGD	Stochastic Gradient Descent
SLFN	Single Layer Forward Network
SVM	Support Vector Machine
TCSC	Thyristor-Controlled Series Capacitor
TD	Temporal Difference
UAV	Unmanned Aerial Vehicle
UVFA	Universal Value Function Approximator

CHAPTER ONE

INTRODUCTION

1.1 RESEARCH BACKGROUND

Most of the robot control and planning issues are represented by uncertainties of sensing and acting devices of robots. Recently, applications in robotics have moved from a highly controlled environment to unstructured environments. Working in an unstructured environment is considered as a challenging task for autonomous agents. The unstructured environment is characterized by one or more of these factors: complexity, uncertainties and/or a high dimensional state space (Katz, Kenney, & Brock, 2008).

Creating an autonomous agent, that gets real observations such as sensory data, images, videos or audio from the environment and learns optimal sequential actions, has been considered as one of the main goals of Artificial General Intelligence (AGI) (Kuhnberger et al., 2009). Deep (Hierarchical) Learning and Reinforcement Learning (RL), two subfields of Machine Learning (ML), can address this objective. Deep learning can learn features that are relevant to discriminative action values. On the other hand, Reinforcement Learning learns to map the learned features to optimal actions (Kuhnberger et al., 2009).

Reinforcement learning (RL) is an active research area in machine learning, artificial intelligence, and neural network (Sutton & Barto, 1998; Kaelbling, Littman, & Moore, 1996). RL differs from other learning methods such as supervised learning that needs to have a pair of inputs/outputs to find the model. In RL, the agent first observes the state of the environment and generates action. The environment moves the

agent to a new state and gives it a reward which is a scalar value that evaluates the action in the current state. The idea of learning by interacting (trial-and-error) with the environment is used when there is no direct teacher but only sensory and motor connections with the surrounding (Sutton & Barto, 1998; Kaelbling, Littman, & Moore, 1996). The objective of RL is to map different situations to proper actions that maximize a reward.

RL has been used for a while to find an optimal policy or optimal series of actions in a low dimensional environment (Sutton & Barto, 1998; Kaelbling, Littman, & Moore, 1996). This policy results from the interaction between the agent and the environment by doing actions and getting positive and negative rewards. The trial – error technique formulates the RL utilizing the MDP (Markov Decision Process) model which describes the agent-environment interaction. Figure 1.1 shows the model of interaction between robot and environment in the unstructured environment. The value iteration method is one of the RL methods that use the value function, which is discounted accumulated rewards, to describe the value of the actions (Bellman, 1957). The objective is to optimize the policy by choosing actions that have maximum action values.

When the input is high dimensional data such as images, traditional RL is suffering from many problems (Bellman, 1957; Keogh & Mueen, 2017). Therefore, the method of deep reinforcement learning was found as an alternative solution. A combination of deep and reinforcement learning is the foundation of AGI (Kuhnberger et al., 2009). Deep reinforcement learning gets a stream of raw data (sensors or camera's images) and reacts to the environment with a sequence of control actions to finally achieve the desired goal. The existing deep reinforcement learning models were found to give a good performance, but they require long training time even if the algorithms run on a powerful Graphical Processing Unit. Therefore, there is still a need to have